



```

sasha@debian:~$ mkdir lab1
sasha@debian:~$ cd lab1
sasha@debian:~/lab1$ echo "данные A" > fileA.txt
sasha@debian:~/lab1$ ln fileA.txt linkA.txt
sasha@debian:~/lab1$ ln -s fileA.txt slinkA.txt
sasha@debian:~/lab1$ ls -li
итого 8
267592 -rw-r--r-- 2 sasha sasha 16 ноя 14 15:38 fileA.txt
267592 -rw-r--r-- 2 sasha sasha 16 ноя 14 15:38 linkA.txt
267533 lrwxrwxrwx 1 sasha sasha 9 ноя 14 15:40 slinkA.txt -> fileA.txt
sasha@debian:~/lab1$ echo "ещё данные" >> fileA.txt
sasha@debian:~/lab1$ cat linkA.txt
данные A
ещё данные
sasha@debian:~/lab1$ cat slinkA.txt
данные A
ещё данные
sasha@debian:~/lab1$ rm fileA.txt
sasha@debian:~/lab1$ ls -li
итого 4
267592 -rw-r--r-- 1 sasha sasha 36 ноя 14 15:42 linkA.txt
267533 lrwxrwxrwx 1 sasha sasha 9 ноя 14 15:40 slinkA.txt -> fileA.txt
sasha@debian:~/lab1$ cat linkA.txt
данные A
ещё данные
sasha@debian:~/lab1$ cat slinkA.txt
cat: slinkA.txt: Нет такого файла или каталога
sasha@debian:~/lab1$ mkdir dir1
sasha@debian:~/lab1$ ln dir1 dir1_hard
ln: dir1: не допускается создавать жёсткие ссылки на каталоги

```

1. Создаём директорию lab1 и перейдите в неё.
2. Создаём файл fileA.txt с текстом данные A.
3. Создаём жёсткую ссылку linkA.txt, указывающую на fileA.txt.
4. Создаём символьную ссылку slinkA.txt, указывающую на fileA.txt.
5. Вывели список файлов в подробном формате с номерами inode: ls -li.
6. Дописали строку ещё данные в fileA.txt.
7. Вывели содержимое linkA.txt.
8. Вывели содержимое slinkA.txt.
9. Удалили fileA.txt.
10. Повторно выполнили ls -li.
11. Вывели содержимое linkA.txt.
12. Вывели содержимое slinkA.txt.
13. Создали директорию dir1.
14. Попытались создать жёсткую ссылку dir1\_hard, указывающую на dir1.

```

sasha@debian:~/lab1$ ln -s dir dir1_soft
sasha@debian:~/lab1$ echo "inner" > dir1/inner.txt
sasha@debian:~/lab1$ ln -s /no/such/file broken1.txt
sasha@debian:~/lab1$ find . -type l
./dir1_soft
./slinkA.txt
./broken1.txt
sasha@debian:~/lab1$ cp slinkA.txt copy1.txt
cp: не удалось выполнить stat для 'slinkA.txt': Нет такого файла или каталога
sasha@debian:~/lab1$ ln linkA.txt linkB.txt
sasha@debian:~/lab1$ rm linkA.txt
sasha@debian:~/lab1$ cat linkB.txt
данные А
ещё данные

```

15. Создали символьную ссылку dir1\_soft, указывающую на dir1.
16. Создали файл dir1/inner.txt.
17. Создали символьную ссылку broken1.txt, указывающую на несуществующий путь /no/such/file.
18. Нашли все символьные ссылки в текущей директории: find . -type l.
19. Скопировали slinkA.txt в copy1.txt (без дополнительных флагов).
20. Создали жёсткую ссылку linkB.txt, указывающую на linkA.txt. Удалите linkA.txt. Проверьте, доступны ли данные через linkB.txt.

### Контрольные вопросы

1. Чем отличается жёсткая ссылка от копии файла?
  - Жёсткая ссылка — это ещё одно имя того же файла. Копия — это отдельный файл со своими собственным данными.
2. Что произойдёт с символьной ссылкой, если удалить целевой файл?
  - Символьная ссылка останется, но будет указывать на несуществующий путь.
3. Что произойдёт с жёсткой ссылкой, если удалить «оригинал»?
  - Ничего: жёсткая ссылка продолжит работать, так как «оригинала» как такового нет — все жёсткие ссылки равноправны. Файл останется существовать, пока хотя бы одна жёсткая ссылка указывает на него.
4. Какой командой создаётся жёсткая и символьная ссылка?
  - Жёсткая ссылка: ln источник ссылка
  - Символьная ссылка: ln -s источник ссылка
5. Как узнать, сколько жёстких ссылок указывает на один и тот же файл? Какой параметр ls это показывает?
  - Количество жёстких ссылок показывает команда ls -l в **третьем столбце** (после прав доступа и владельца файла).
6. Может ли символьная ссылка ссылаться на несуществующий путь? А жёсткая?
  - Символьная ссылка может указывать на несуществующий путь.
  - Жёсткая ссылка, её можно создать только на существующий путь.
7. Как жёсткие ссылки помогают экономить место при резервном копировании?
  - Жёсткие ссылки позволяют нескольким каталогам «делить» один и тот же набор данных без копирования. Поэтому резервные копии могут создавать новые жёсткие ссылки, а не дублировать файлы, экономя место на диске.