**Как правильно создавать символические ссылки:**

27 прочтений

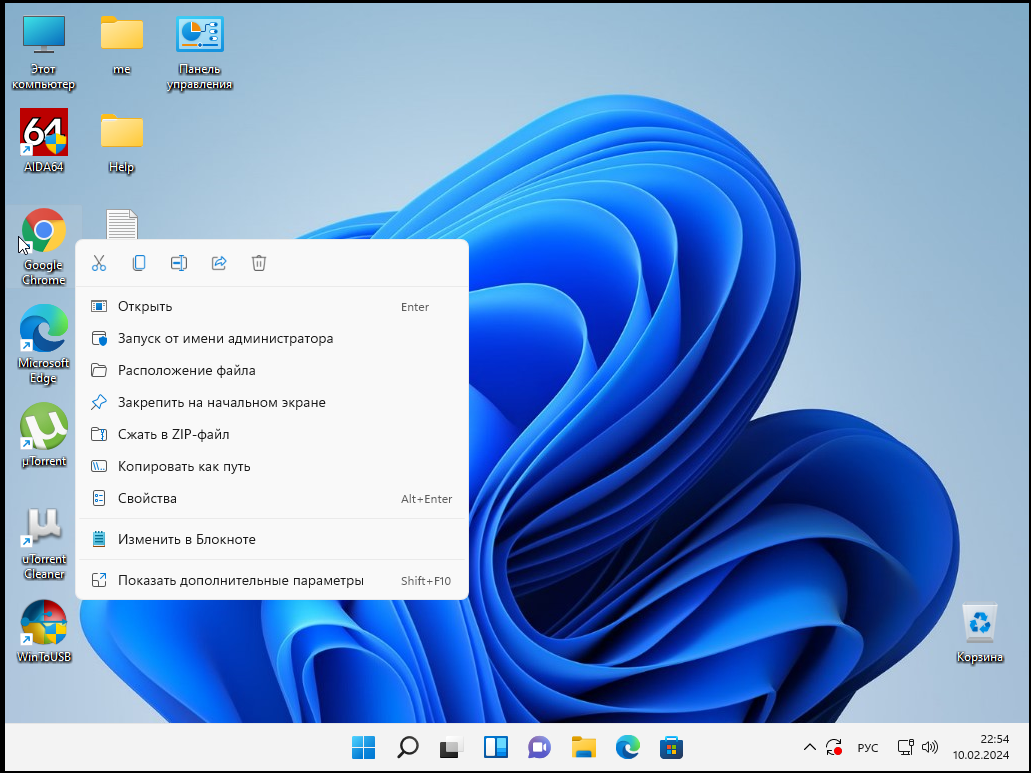
10 февраля

Символические ссылки часто сравнивают с ярлыками в Windows.

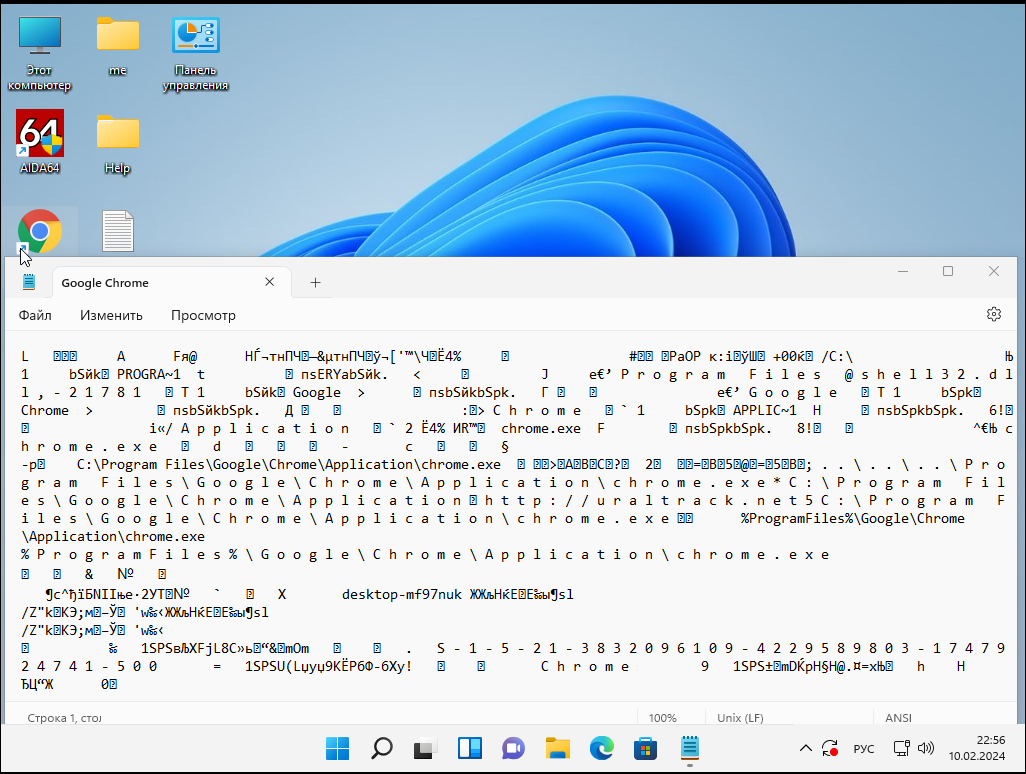
В этом действительно есть некоторое сходство. Если кликнуть по ярлыку в Windows, то будет запущена программа, или открыт документ, на который ярлык ссылается.

Так если мы нажмем на ярлык Google Chrome на рабочем столе Windows, будет запущен браузер.

Чисто технически ярлык это файл, в котором записан путь на объект, на который ярлык ссылается.



Но если попытаться открыть этот файл в текстовом документе (например, через контекстное меню "Изменить в Блокноте", то будет открыт не тот файл, на который ярлык ссылается, а именно сам файл, хранящий путь к объекту.



То есть ярлык в Windows обрабатывается не операционной системой, а приложением-проводником, реализующим интерфейс рабочего стола.

Все по другому с символическими ссылками в GNU/Linux.

Если создать символическую ссылку на приложение, и вызвать ее на выполнение (не важно, через графическое приложение, как в Windows, или набрав ее имя в командной строке), будет вызвано то приложение, на которое создана символическая ссылка.

Но если открыть на просмотр эту ссылку, то мы увидим не адрес, на который ссылка ссылается, а сам код программы. Нам повезет, если это скрипт, а если это бинарный исполняемый файл, мы увидим кракозябры.

*Далее упражнение можно повторять за мной (только учтите, что вашего пользователя зовут не me, а как-то по-другому).*

Попробуем создать символическую ссылку.

Перейдем в домашний каталог с помощью команды cd и стандартного для linux обозначения домашнего каталога «~»

cd ~

-3

Создадим в домашнем каталоге две папки test1 и test2 с помощью команды mkdir

mkdir test1 test2

-4

Напомню, что mkdir может принимать несколько аргументов, для создания нескольких каталогов, чем мы и пользуемся.  
И также напомню, что папка, каталог и директория — синонимы.

Перейдем в каталог test1 с помощью команды cd

cd test1

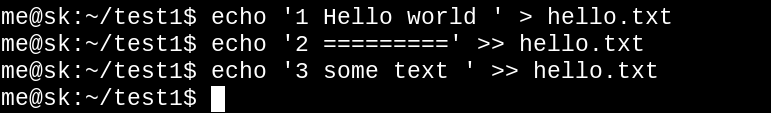
-5

Создадим в каталоге test1 файл hello.txt с помощью команды echo и перенаправления.

echo '1 Hello world ' > hello.txt

echo '2 =========' >> hello.txt

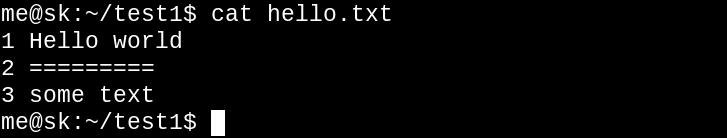
echo '3 some text ' >> hello.txt



Напомню, что по умолчанию перенаправление «>» перезаписывает файл, а «>>» добавляет в конец.

Посмотрим содержимое получившегося файла с помощью команды cat

cat hello.txt



Теперь перейдем в каталог test2.

Это можно сделать двумя способами.

Перейти на уровень выше, используя обозначение родительского каталога «..»

cd ..

-8

И сразу перейти в каталог test2

cd test2

-9

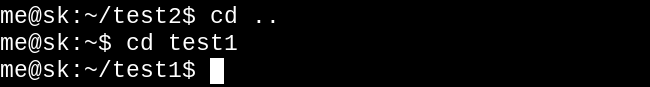
Команда pwd (print working directory) печатает полный путь для текущего каталога.

-10

Чтобы продемонстрировать второй способ, вернемся в каталог test1.

cd ..

cd test1



Убедимся, что мы действительно в test1 с помощью команды pwd

-12

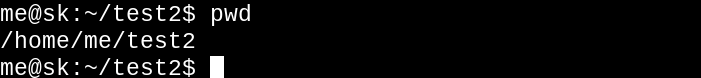
А теперь вернемся из test1 в test2 одной командой

cd ../test2

-13

cd не понимает нескольких аргументов, но зато можно использовать относительный путь, содержащий столько подъемов наверх (..) и наоборот вхождений в подкаталоги, сколько надо.

И с помощью pwd мы убеждаемся, что снова в test2



Как нам посмотреть содержимое файла hello.txt, находящегося в каталоге test1, не выходя их каталога test2?

Правильно, указав относительный путь. На один уровень вверх (..), далее в каталоге test1 лежит файл hello.txt

cat ../test1/hello.txt



Итак, этот путь работает.  
А значит, мы можем создать на него символическую ссылку с помощью команды ln -s

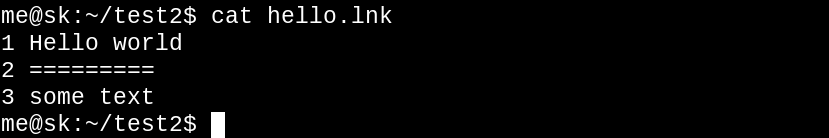
ln -s ../test1/hello.txt hello.lnk

-16

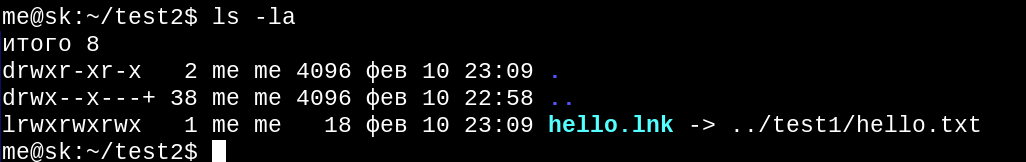
*Только не путайте команды ln и ls, это разные команды ;)*

Проверяем, что все получилось

cat hello.lnk



Посмотрим, как отображается в перечне файлов в выводе ls подробном (используем ключ ls -la)

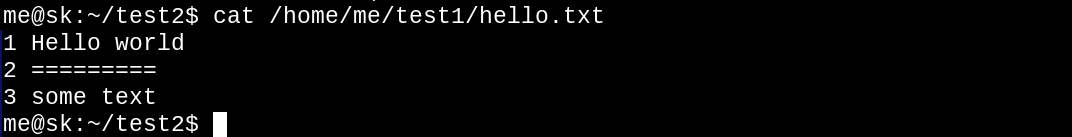


Другой способ создать ссылку — использовать абсолютный путь.

*Моего пользователя зовут me, и его домашний каталог /home/me (но ни в коем случае не /home!*  
*/home — это каталог, в котором хранятся домашние каталоги пользователей, но не сам домашний каталог).*  
*Если вашего ползователя зовут user, то ваш домашний каталог будет /home/user, если вы вошли этим пользователем в систему.*  
*А если ваш пользователь — john, то его домашний каталог будет /home/john.*  
*А вот для пользователя root домашний каталог — /root, но не нужно работать рутом!*

Проверяю что файл доступен по абсолютному пути

cat /home/me/test1/hello.txt



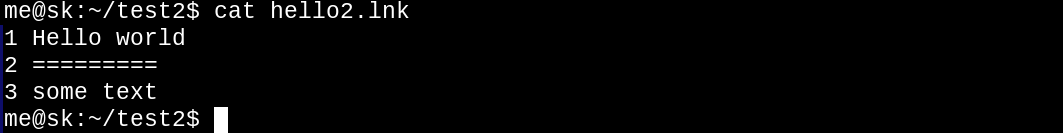
Создаем ссылку

ln -s /home/me/test1/hello.txt hello2.lnk

-20

Проверяем, что все работает

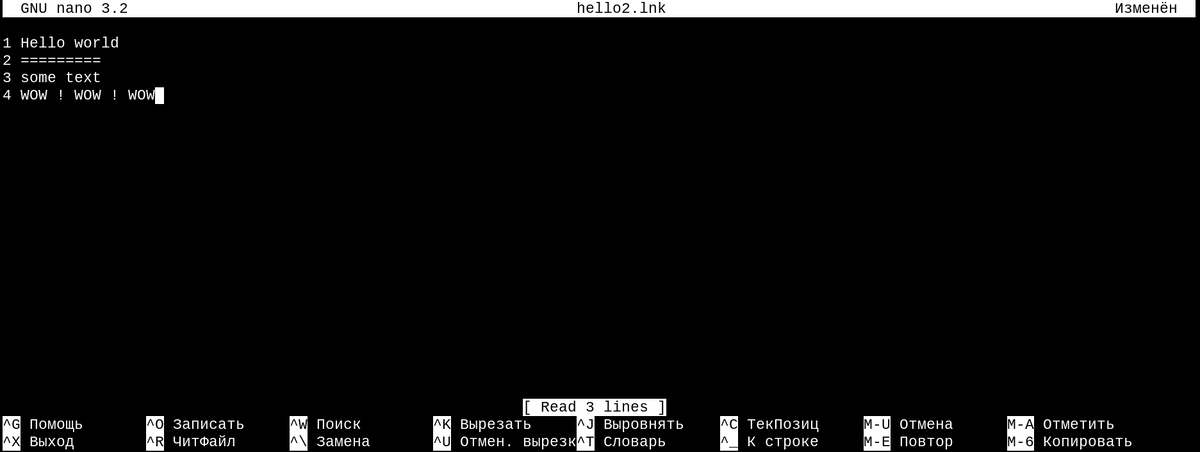
cat hello2.lnk



Мы можем даже отредактировать файл, работая со ссылкой

nano hello2.lnk

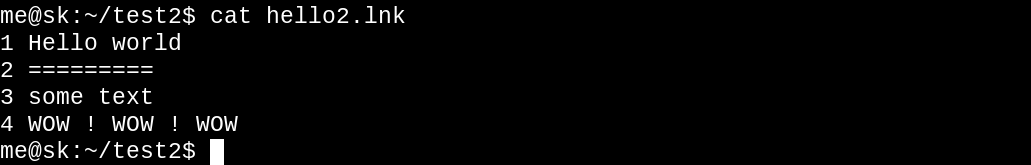
-22



Сохраняем (F2)

и проверяем, что файл изменен

cat hello2.lnk

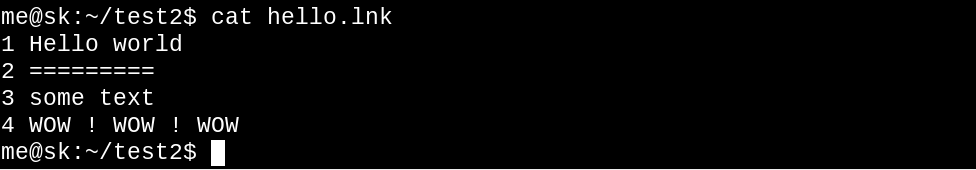


И убедимся, что на самом деле изменен был исходный файл

cat ../test1/hello.txt

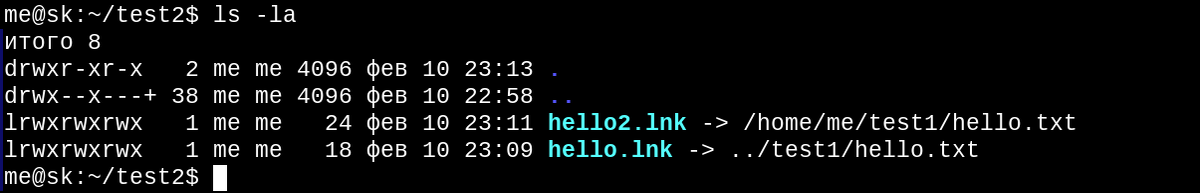
По первой созданной ссылке тоже отображается тот же самый файл

cat hello.lnk



Посмотрим на ссылки в списке файлов

ls -la



Обратите внимание, каким цветом отображаются ссылки.

Ссылки будут рабочими, пока существует файл по пути, указанном в ссылке.  
Если файл был удален, переименован, перемещен, или носитель, на котором он находился, будет отмонтирован, тогда ссылка станет не рабочей, битой.

Давайте переименуем исходный файл с помощью команды mv

mv ../test1/hello.txt ../test1/hello.old

-27

Посмотрим, удается ли файл просмотреть с помощью ссылки?

cat hello.lnk



Не удается, пути, куда ссылка ссылается нет.

Посмотрим, как отображаются ссылки в перечне файлов

ls -la



Ссылки отображаются красным, они битые.

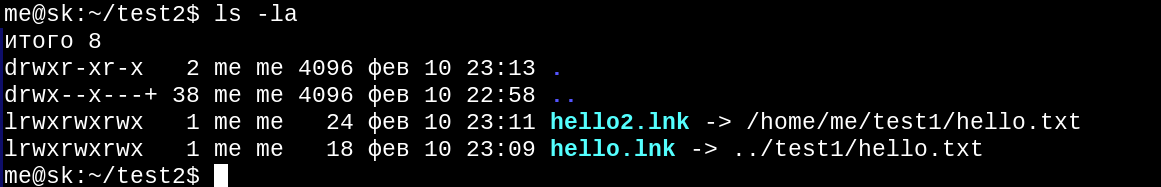
Вернем исходный файл

mv ../test1/hello.old ../test1/hello.txt

-30

Смотрим, восстановились ли ссылки?

ls -la



И ссылки снова рабочие.

Обратите внимание, что работа с просмотром и редактированием — это манипуляции с файлом, на который ссылка ссылается.

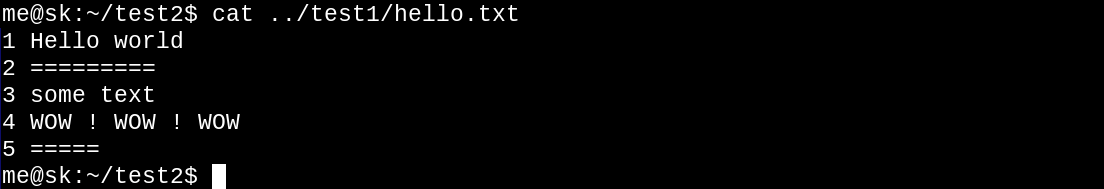
Добавим одну строку, используя перенаправление (>>) и имя ссылки.

echo '5 ====='>>hello.lnk

-32

А посмотрим содержимое исходного файла

cat ../test1/hello.txt



А вот переименование или удаление ссылки — это работа с самой ссылкой.

mv hello.lnk hello.link

-34

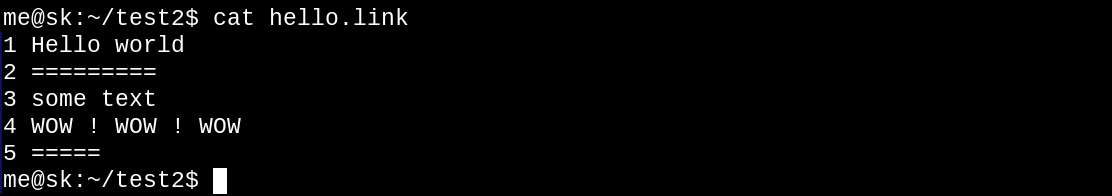
На исходный файл это действие не повлияло — исходный файл на месте

cat ../test1/hello.txt



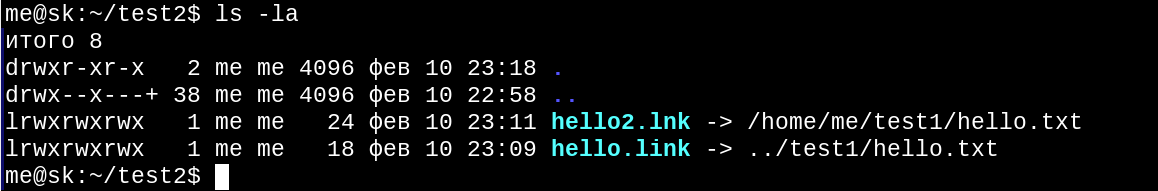
По новому имени ссылки тоже отзывается

cat hello.link



Посмотрим все вместе

ls -la



Удаление ссылки -- это тоже удаление только ссылки

rm hello2.lnk

-38

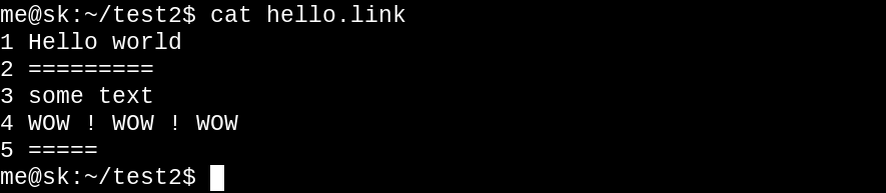
Исходный файл не пострадал

cat ../test1/hello.txt

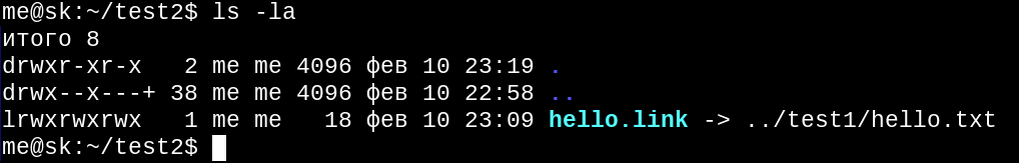


Вторая ссылка по прежнему работает

cat hello.link



Ну и ls -la напоследок текущего каталога



И каталога test1 для сравнения

ls -a ../test1

