

Wiemy już jak tworzyć twarde dowiązania do zwykłych plików. A co z katalogami i pozostałymi plikami specjalnymi? Poza dowiązaniem twardego są jeszcze **dowiązania symboliczne**, albo linki symboliczne (symbolic links), dla odróżnienia od twardego linków zwane czasem miękkimi linkami (soft links). Tworzymy je tym samym poleceniem `ln` jak twarde, ale dodajemy opcję `-s`.

```
sirius$ ln -s cosik.txt doc.txt
sirius$ ls -li cosik.txt doc.txt
 12901 -rw-r--r--  2 mariusz  staff    23 lis 29 17:17 cosik.txt
 14292 lrwxrwxrwx  1 mariusz  staff     9 lis 29 17:34 doc.txt -> cosik.txt
```

Powstaje w ten sposób plik specjalny, jakiego widzieliśmy wcześniej o symbolu `l` jako typ pliku z charakterystyczną strzałką w ostatniej kolumnie. Teraz znaczenie tej strzałki staje się oczywiste i podpowiada o co chodzi.

Zwróćmy uwagę, że powstał zupełnie nowy plik o numerze 14292. O tyle dziwne, że oryginał zajmuje 23 bajty podczas, gdy dowiązanie symboliczne tylko 9 bajtów. Policzmy z ilu znaków składa się nazwa `cosik.txt`. Liczba 9 to długość nazwy oryginału i rozmiar linku symbolicznego. Koincydencja nie jest przypadkowa, bo dowiązanie symboliczne to odrębny plik, o nowym numerze inode, zawierającym ścieżkę do oryginału, taką jak podano przy tworzeniu.

```
sirius$ ln -s test1/test2/nic.txt doc2.txt
sirius$ ls -li cosik.txt doc.txt doc2.txt
 12901 -rw-r--r--  2 mariusz  staff    23 lis 29 17:17 cosik.txt
 14292 lrwxrwxrwx  1 mariusz  staff     9 lis 29 17:34 doc.txt -> cosik.txt
 17540 lrwxrwxrwx  1 mariusz  staff    19 lis 29 17:40 doc2.txt ->
test1/test2/nic.txt
```

Tym razem link o nazwie `doc2.txt` zajmuje 19 bajtów tyle co ścieżka względna `test1/test2/nic.txt`.

Podobnie jak z linkami twarde, każda operacja na linku symbolicznym zostanie wykonana na oryginale. W trakcie, gdy używamy linku symbolicznego, system operacyjny podstawia inode oryginału. Nasuwa się pytanie, czy zawsze może to zrobić? Co się stanie, gdy usuniemy oryginał?

```
sirius$ cat doc2.txt
I hate UFS
UFS is cool
sirius$ rm test1/test2/nic.txt
sirius$ cat doc2.txt
cat: cannot open doc2.txt
sirius$ ls -l doc2.txt
lrwxrwxrwx  1 mariusz  staff    19 lis 29 17:40 doc2.txt ->
test1/test2/nic.txt
```

Pomimo, że usunęliśmy oryginał dowiązanie zostało, ale każde odwołanie do niego daje efekt taki jakbyśmy odwoływali się do nieistniejącego pliku. Może to wydawać się bez sensu, ale czasem jest wygodne. Możemy mieć link do katalogu, który pojawi się dopiero po zamontowaniu odpowiedniego systemu plików.

Kiedy dowiązania symboliczne mogą się przydać? Gdy na przykład mamy ileś ponumerowanych wersji pliku z kodem źródłowym: `ver1.c`, `ver2.c`, `ver3.c`, aby nie musieć pamiętać jaki numer jest ostatni, podczas tworzenia nowej wersji jednocześnie tworzymy dowiązanie symboliczne

```
current.c -> ver3.c.
```

Dowiązania symboliczne możemy tworzyć do katalogów.

```
sirius$ ln -s test1 testx
sirius$ ls -ldi test1 testx
 29141 drwxr-xr-x  3 mariusz  staff    512 lis 29 16:42 test1
 17542 lrwxrwxrwx  1 mariusz  staff     5 lis 29 17:56 testx -> test1
```

Warto zapamiętać opcję `-L` do `ls` aby zamiast linku zobaczyć oryginał:

```
sirius$ ls -lL testx
total 2
drwxr-xr-x  2 mariusz  staff    512 lis 29 17:45 test2
```

Akurat dla katalogu to samo uzyskamy dopisując slash na końcu:

```
sirius$ ls -l testx/
total 2
drwxr-xr-x  2 mariusz  staff    512 lis 29 17:45 test2
```

Dowiązanie symboliczne do katalogu zachowuje się jak katalog oryginalny:

```
sirius$ cd testx
sirius$ ls -l
total 2
drwxr-xr-x  2 mariusz  staff    512 lis 29 17:45 test2
```