

# Linki symboliczne i twarde w systemach Linux/Unix

## Wprowadzenie:

Linki w systemach Linux pozwalają na tworzenie alternatywnych nazw i dróg dostępu do plików bez duplikowania ich zawartości. Są dwa typy linków: symboliczne (symlinki) i twarde (hard links), każdy z nich ma swoje zastosowania i ograniczenia.

## 1. Link symboliczny (symbolic link, symlink)

### Co to jest link symboliczny?

Link symboliczny to specjalny plik, który zawiera ścieżkę do innego pliku lub katalogu. Można go traktować jak "skrót" w systemach Windows. Link wskazuje na lokalizację docelową, ale sam nie zawiera danych oryginalnego pliku.

### Tworzenie linku symbolicznego:

```
In -s /ścieżka/do/originalu /ścieżka/do/linku
```

### Przykłady praktyczne:

```
In -s /usr/local/bin/python3.11 /usr/local/bin/python
```

Tworzy link `python` wskazujący na `python3.11`. Dzięki temu można wywoływać `python` zamiast pełnej nazwy.

```
In -s /var/www/html/projekty/strona ~/strona_link
```

Tworzy wygodny skrót do katalogu projektu w katalogu domowym.

```
In -s /mnt/dysk_zewnetrzny/backup ~/backup
```

Skrót do katalogu backupu na zewnętrznym dysku.

### Sprawdzanie linków symbolicznych:

```
ls -l link_nazwa
```

Wyświetla informację o linku i pokazuje gdzie wskazuje, np.:

```
lrwxrwxrwx 1 user user 24 Nov 23 15:00 strona_link -> /var/www/html/projekty/strona
```

### Usuwanie linku symbolicznego:

```
rm link_nazwa
```

Uwaga: usuwanie linku nie usuwa pliku docelowego!

### Cechy linków symbolicznych:

- Mogą wskazywać na pliki i katalogi
- Mogą wskazywać na nieistniejące cele (tzw. "wiszące linki")
- Działają między różnymi systemami plików i partycjami
- Są widoczne jako osobny typ pliku (oznaczenie `l` w `ls -l`)

## 2. Link twardy (hard link)

### Co to jest link twardy?

Link twardy to dodatkowa nazwa dla tego samego pliku na dysku. W przeciwieństwie do linku symbolicznego, link twardy wskazuje bezpośrednio na te same dane (ten sam inode) co oryginalny plik. W praktyce nie ma różnicy między "oryginałem" a "linkiem" - oba są równorzędnymi nazwami tego samego pliku.

### Tworzenie linku twardego:

```
ln /ścieżka/do/pliku /ścieżka/do/twardego_linku
```

### Przykłady praktyczne:

```
ln /var/log/syslog /home/admin/syslog_backup
```

Tworzy drugi dostęp do tego samego pliku logów. Modyfikacja w jednym miejscu jest widoczna w drugim.

```
In dokument.txt dokument_kopia.txt
```

Obie nazwy wskazują na ten sam plik. Usunięcie jednej nazwy nie usuwa danych, dopóki istnieje druga.

### Sprawdzanie linków twardych:

```
ls -li plik
```

Opcja `-i` pokazuje numer inode. Pliki o tym samym numerze inode to linki twarde do tych samych danych.

```
$ ls -li dokument.txt dokument_kopia.txt
12345678 -rw-r--r-- 2 user user 1024 Nov 23 15:00 dokument.txt
12345678 -rw-r--r-- 2 user user 1024 Nov 23 15:00 dokument_kopia.txt
```

Ten sam numer inode (12345678) i liczba linków "2" oznacza, że to linki twarde.

### Cechy linków twardych:

- Działają tylko dla plików (nie dla katalogów)
- Muszą być w tym samym systemie plików
- Plik jest usuwany dopiero gdy usuniemy wszystkie linki twarde do niego
- Nie są widoczne jako osobny typ - wyglądają jak zwykłe pliki

## 3. Różnice między linkami symbolicznymi a twardymi

Cecha	Link symboliczny	Link twardy
Typ	Wskaźnik do ścieżki	Dodatkowa nazwa pliku
Katalogi	Tak	Nie
Różne systemy plików	Tak	Nie
Po usunięciu oryginału	Link "wiszący"	Dane nadal dostępne

## 4. Praktyczne zastosowania

```
# Link symboliczny do aktualnej wersji aplikacji
ln -s /opt/app-1.2.3 /opt/app

# Link symboliczny do konfiguracji
ln -s /etc/nginx/sites-available/strona.conf /etc/nginx/sites-enabled/strona.conf

# Link twardy jako backup przed edycją
ln waźny_plik.conf waźny_plik.conf.backup

# Sprawdzenie gdzie prowadzi link
readlink -f link_nazwa

# Znalezienie wszystkich linków twardych do pliku
find / -inum $(ls -i plik | awk '{print $1}') 2>/dev/null
```

---

Revision #1

Created 23 November 2025 14:19:07 by Tomasz Konieczny

Updated 23 November 2025 14:20:01 by Tomasz Konieczny