

Klasa: III TI Technikum Kształtowania Środowiska - Technik Informatyk

Systemy operacyjne.

Temat: Dowiązania do plików.

Wykonałam zrzuty z Podręcznika: WSiP „Systemy operacyjne i sieci komputerowe cz2.”  
K.Pytel, S.Osetek. Proszę zapoznać się z treścią tego podręcznika.

Przeczytaj rozdział dotyczący dzisiejszej lekcji.

### 6.4.2. Dowiązania do plików

Linux przechowuje informacje o zbiorach, np. plikach lub katalogach, w strukturach nazywanych i-węzłami (*i-mode*). Każdy plik ma 1 węzeł. Identyfikacja pliku odbywa się na podstawie jego numeru, unikatowego w obrębie danego systemu plików. Człowiek woli posługiwać się nazwami plików, które również są skojarzone z odpowiednimi i-węzłami. Dowiązania (*links*) umożliwiają odwoływanie się do jednego pliku za pomocą różnych nazw, pozwalają również na umieszczenie jednego pliku w wielu miejscach w strukturze plików.

Dowiązania dzielą się na twarde (*hard links*) i symboliczne (*symbolic links*). **Dowiązanie twarde** jest to referencja wskazująca konkretny, istniejący wcześniej i-węzeł w obrębie tej samej partycji (systemu plików). Dla systemu operacyjnego dowiązanie takie jest po prostu dodatkową nazwą wskazywanego obiektu. Plik, mający *n* dowiązań ma też *n* nazw. Aby skasować obiekt w systemie plików, trzeba usunąć wszystkie odwołujące się do niego dowiązania.

Do tworzenia dowiązań twardej służy polecenie:

```
ln cel_dowiązania dowiązanie
```

gdzie:

`cel_dowiązania` – plik, do którego chcemy zrobić dowiązanie  
`dowiązanie` – dodatkowa nazwa pliku, równoprawna z wcześniej utworzonymi nazwami.

#### Przykład

```
ln /home/uczen/plik.txt /root/plik_ucznia.txt
```

**Dowiązanie symboliczne** to skojarzenie nowej nazwy z istniejącą wcześniej nazwą zbioru (nie bezpośrednio z i-węzłem). Dowiązanie symboliczne wskazuje na nazwę pliku lub katalogu, która dopiero wskazuje na i-węzeł. Odpowiednikiem dowiązania symbolicznego w systemie Windows jest skrót.

Dowiązania symboliczne tworzy się analogicznie, jak dowiązania twarde, tylko dodając do polecenia `ln` parametr `-s`, np.

```
ln -s /home/uczen/plik.txt /root/plik_ucznia.txt
```

Dowiązania kasujemy tak jak pliki. Przykłady użycia poleceń związanych z tworzeniem i usuwaniem dowiązań pokazano na rys. 6.2.

```

root@kpfcs:~/linki
plik Edycja Widok Terminal Karty Pomoc
[root@kpfcs linki]# rm link_symboliczny
[root@kpfcs linki]# ln /home/student1/plik.txt link
[root@kpfcs linki]# ln -s /home/student1/plik.txt link_symboliczny
[root@kpfcs linki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:27 .
drwx----- 14 student1 kp 4096 lip 18 21:26 ..
-rw-r--r-- 2 root root 0 lip 18 21:24 link
lrwxrwxrwx 1 root root 28 lip 18 21:27 link_symboliczny -> /home/student1/plik.txt
[root@kpfcs linki]# rm link
[root@kpfcs linki]# rm link_symboliczny
[root@kpfcs linki]#

```

Rys. 6.2. Polecenia do tworzenia i usuwania dowiązań

### 6.4.3. Błędy związane z dowiązaniem

Dowiązanie twarde wskazuje na miejsce na dysku. Jeżeli plik skasujemy i w jego miejscu umieścimy inny plik, to nasze dowiązanie będzie wskazywało na to samo miejsce, ale zupełnie inny plik.

Dowiązanie symboliczne jest związane z nazwą pliku. Na rysunku 6.3 pokazano dowiązanie symboliczne `link_s` utworzone do pliku `plik.txt`. Jeżeli skasujemy plik `.txt`, do którego prowadzi dowiązanie, to system wyświetli błąd. Dowiązanie istnieje, ale prowadzi do nieistniejącego pliku. Jeżeli utworzymy plik o takiej samej nazwie, to dowiązanie będzie wskazywać nowy plik, mimo że jego zawartość może być zupełnie inna.

```

root@kpfcs:~/linki
plik Edycja Widok Terminal Karty Pomoc
[root@kpfcs linki]# touch plik.txt
[root@kpfcs linki]# ln -s plik.txt link_s
[root@kpfcs linki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:31 .
drwx----- 14 student1 kp 4096 lip 18 21:26 ..
lrwxrwxrwx 1 root root 8 lip 18 21:31 link_s -> plik.txt
-rw-r--r-- 1 root root 0 lip 18 21:30 plik.txt
[root@kpfcs linki]# rm plik.txt
[root@kpfcs linki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:32 .
drwx----- 14 student1 kp 4096 lip 18 21:26 ..
lrwxrwxrwx 1 root root 8 lip 18 21:31 link_s -> plik.txt
-rw-r--r-- 1 root root 0 lip 18 21:32 plik.txt
[root@kpfcs linki]# touch plik.txt
[root@kpfcs linki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:32 .
drwx----- 14 student1 kp 4096 lip 18 21:26 ..
lrwxrwxrwx 1 root root 8 lip 18 21:31 link_s -> plik.txt
-rw-r--r-- 1 root root 0 lip 18 21:32 plik.txt
[root@kpfcs linki]#

```

Rys. 6.3. Błędy związane z dowiązaniem

Temat: Uprawnienia do plików i folderów.

Wykonałam zrzuty z Podręcznika: WSiP „Systemy operacyjne i sieci komputerowe cz2.” K.Pytel, S.Osetek. Proszę zapoznać się z treścią tego podręcznika.

Przeczytaj rozdział dotyczący dzisiejszej lekcji.

### 6.4.6. Uprawnienia do plików i folderów

Każdy użytkownik niebędący administratorem Linuksa ma dostęp tylko do niektórych plików. Dzięki temu nikt nie może np. zmodyfikować plików należących do innego użytkownika. Do każdego pliku jest przypisany **identyfikator właściciela** `u` (user) – użytkownika, który stworzył ten plik, oraz **grupy** `g` (group) – czyli zbioru użytkowników, którzy mają do tego pliku uprawnienia, w przeciwieństwie do **pozostałych** `o` (others).

Istnieją trzy podstawowe prawa dostępu do pliku: `r` – prawo do **odczytu**, `w` – prawo do **zapisu**, `x` – prawo do **uruchomienia**, jeśli plik jest programem. Prawa te są nadawane niezależnie właścicielowi pliku, grupie, do której plik należy, i pozostałym użytkownikom. Aby dowiedzieć się, jakie uprawnienia są ustawione dla poszczególnych plików, używa się komendy `ls -l`. Każdy plik i folder w systemie Linux posiada 10 bitów protekcji w formacie `drwxrwxrwx`, gdzie:

- bit 1 – identyfikacja rodzaju pliku (`d` – folder, `-` – plik, `l` – link do pliku),
- bity 2–4 – uprawnienia właściciela pliku,
- bity 5–7 – uprawnienia grupy, do której należy właściciel,
- bity 8–10 – uprawnienia pozostałych użytkowników.

Litera oznacza ustawione uprawnienie, kreska brak uprawnienia.

Administrowanie uprawnieniami możliwe jest w oparciu o polecenie `chmod`, np.

```
chmod ugo+w plik.txt
```

powoduje dodanie (`+`) prawa do zapisu (`w` (write)) dla pliku o nazwie `plik.txt` właścicielowi (`u` (user), użytkownikom należącym do tej samej grupy, co właściciel pliku `g` (group) oraz dodanie prawa do zapisu pliku wszystkim pozostałym użytkownikom `o` (others).

Każde z praw dostępu ma przypisany odpowiedni parametr cyfrowy:

<code>r</code>	prawa do odczytu	4
<code>w</code>	prawa do zapisu	2
<code>x</code>	prawa do uruchomienia	1
<code>-</code>	brak praw dostępu	0

Dodając do siebie odpowiednie parametry, zestaw trzech praw możemy przedstawić za pomocą jednej cyfry.

Oto kilka najczęściej spotykanych kombinacji:

<code>---</code>	0	brak praw
<code>---</code>	4	prawa do zapisu
<code>rw-</code>	6	prawa do zapisu i odczytu
<code>rx-</code>	7	prawa do zapisu, odczytu i uruchomienia
<code>r-x</code>	5	prawa do odczytu i uruchomienia
<code>-x-</code>	1	prawa do uruchomienia

Do zmiany uprawnień możemy wykorzystać również polecenie `chmod` oraz uprawnienń zapisanych jako 3-cyfrowa liczba. Cyfry od lewej oznaczają uprawnienia dla właściciela, grupy i pozostałych użytkowników. Aby ustawić uprawnienia, należy wydać polecenie

```
chmod 750 plik.txt
```

Przypisane zostaną uprawnienia 7 dla właściciela (`rwx`), 5 dla grupy (`r-x`), 0 dla pozostałych użytkowników (`---`).

### 6.4.7. Właściciele zbiorów

Właścicielem każdego pliku i folderu jest użytkownik i grupa użytkowników. Przenieść własność zbioru na innego użytkownika może tylko administrator. Do zmiany właściciela używamy polecenia, którego składnia jest następująca

`chown użytkownik:grupa plik` (rys. 6.5).



```

root@pfcs:~/www/student1/piki
ls [dajca Wlasc Dmniał Karty Panoł
[roo@pfcs piki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:37 .
drwx----- 5 student1 student1 4096 lip 18 21:37 ..
-rw-r--r-- 1 root root 0 lip 18 21:37 plik.txt
[roo@pfcs piki]# chown student1:student1 plik.txt
[roo@pfcs piki]# ls -la
razem 8
drwxr-xr-x 2 root root 4096 lip 18 21:37 .
drwx----- 5 student1 student1 4096 lip 18 21:37 ..
-rw-r--r-- 1 student1 student1 0 lip 18 21:37 plik.txt
[roo@pfcs piki]#

```

Rys. 6.5. Przenoszenie własności pliku na innego użytkownika

#### Uwaga

Zmiana właściciela zbioru może spowodować zmianę uprawnień przypisanych użytkownikom do danego zbioru.

Pozdrawiam 3T1

Bogusława Kocałek