dn01.04.2020r.

Nauczyciel: Bogusława Kocałek

Klasa: II TI Technikum Kształtowania Środowiska - Technik Informatyk

SYSTEMY OPERACYJNE

Temat: Zarządzanie plikami i folderami.

Przeczytaj rozdział dotyczący dzisiejszej lekcji.

## 6.4. Zarządzanie plikami i folderami

Do wygodnego zarządzania plikami i folderami wykorzystać możemy programy narzędziowe, np. *Midnight Commander* lub środowisko graficzne. Oprócz tego dysponujemy poleceniami umożliwiającymi wykonanie wszystkich zadań z poziomu konsoli. Najczęściej używane polecenia do zarządzania plikami:

- `mkdir` – tworzy folder, np. `mkdir uczen`,
- `rmdir` – usuwa pusty folder, np. `rmdir /uczen`, (sprawdź w pomocy, jak usunąć niepusty folder),
- `cp` – kopiuje pliki, np. `cp /home/uczen1/plik.txt /home/uczen2/`,
- `mv` – przenosi pliki, np. `mv /home/uczen1/plik.txt /home/uczen2`,
- `rm` – usuwa pliki, np. `rm stare.txt`,
- `cd` – zmienia aktualny folder roboczy, np. `cd /`,
- `pwd` – wyświetla ścieżkę aktualnego folderu roboczego,
- `ls` – listuje pliki z folderu, np. `ls`, najczęściej używać będziemy z opcją `-ls` (sprawdź w pomocy, co to zmieni w działaniu polecenia),
- `find` – wyszukuje np. `find /muzyka -name uczen`,
- `touch` – tworzy plik tekstowy, np. `touch plik.txt`,
- `ln` – tworzy nowe dowiązanie do pliku, np. `ln -s link`,
- `rename` – umożliwia zmianę nazwy pliku,
- `cat` – wyświetla zawartość pliku tekstowego.

Większość z powyższych poleceń działa analogicznie jak w systemie Windows. Omówienia wymagają jednak polecenia `find` i `ln`.

### 6.4.1. Wyszukiwanie plików

Polecenie `find` służy do wyszukiwania plików. Jeżeli chcemy znaleźć plik **tekst.txt** w folderze `/home/student` wystarczy napisać:

```
find /home/student -name tekst.txt.
```





Pierwszy argument oznacza przeszukiwany folder wraz z podfolderami, drugi parametr określa warunek, np. `-name` oznacza, że chcemy szukać pliku o określonej nazwie (sprawdź w pomocy, jak znaleźć wszystkie pliki, których jesteś właścicielem). Polecenie `find` akceptuje również nazwy plików określone za pomocą znaków globalnych, np. `find /home/uczen -name *.bat`

### 6.4.2. Dowiązania do plików

Linux przechowuje informacje o zbiorach, np. plikach lub katalogach, w strukturach nazywanych i-węzłami (*i-node*). 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 twardych 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
```

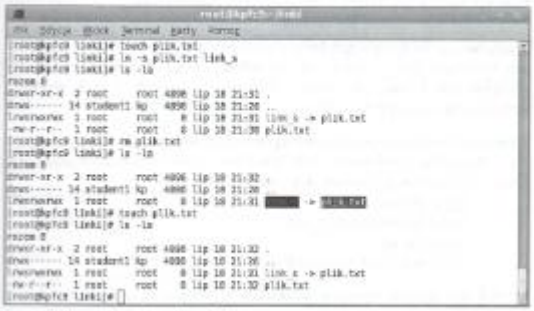
Dowiązania kasujemy tak jak pliki. Przykłady użycia poleceń związanych z tworzeniem i usuwaniem dowiązań pokazano na rys. 6.2.

Rys. 6.2. Polecenia do tworzenia i usuwania dowiązań

## 6.4.3. Błędy związane z dowiązaniami

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.



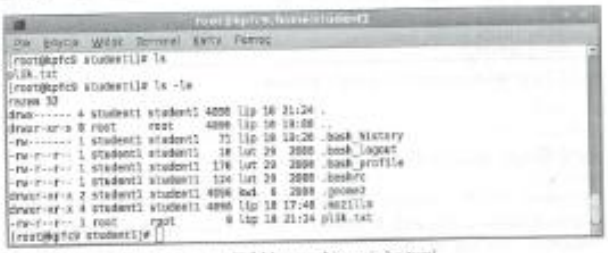
Rys. 6.3. Błędy związane z dowiązaniami





## 6.4.4. Zbiory ukryte

Jeżeli plik lub folder ma być ukryty przed użytkownikiem, to jego nazwa powinna rozpoczynać się od kropki. Aby wyświetlić zbiory ukryte, należy użyć polecenia `ls` z opcją `-a`. Przykład działania polecenia listującego zawartość folderu ze zbiorami ukrytymi pokazano na (rys. 6.4).



Rys. 6.4. Polecenie listujące zawartość folderu ze zbiorami ukrytymi

## 6.4.5. Ścieżki dostępu

Podobnie jak w systemie Windows, w Linuksie do odnajdywania zbiorów na dysku posługujemy się **ścieżkami dostępu**. Ścieżki mogą być bezwzględne lub względne.

**Bezwzględna** ścieżka dostępu rozpoczyna się od folderu głównego, pierwszym znakiem ścieżki bezwzględnej jest ukośnik `/`.

**Względna** ścieżka dostępu przedstawia lokalizację pliku lub folderu względem folderu bieżącego. Jeśli ścieżka dostępu nie rozpoczyna się ukośnikiem, jest ścieżką względną. Aby z folderu bieżącego przejść niżej w strukturze drzewa, wystarczy wpisać ścieżkę, rozpoczynając od nazwy następnego folderu.

Foldery bardzo często wykorzystywane przez użytkowników Linuksa posiadają specjalne oznaczenia umożliwiające szybkie przemieszczanie się pomiędzy nimi. Lokalizacja tych folderów jest zawsze określona w systemie:

- `/` – folder główny,
- `~` – folder domowy użytkownika,
- `.` – folder bieżący,
- `..` – folder bezpośrednio nadrzędny.

Symboli tych możemy używać w ścieżkach dostępu, np.

```
cp ~/skrypty/konta.bat /root
```



Pozdrawiam 2TI

Bogusława Kocałek