

Administrowanie systemami sieciowymi

Sprawozdanie z laboratorium

Data	Tytuł zajęć	Uczestnicy
09.12.2020 11:15	System plików	Bartosz Rodziewicz (226105)

Opis środowiska

Zajęcia laboratoryjne z części nt. systemu Linux zostały wykonane na maszynie wirtualnej postawionej z wykorzystaniem VirtualBox. Zainstalowana w maszynie dystrybucja to Manjaro 20.2 ze środowiskiem (DE) KDE Plasma. Do zajęć użyta została czysta instalacja systemu po doinstalowaniu najnowszych aktualizacji pakietów. W wielu miejscach używany jest alias `ll`, który jest aliasem `ls -aIf`.

Przebieg laboratorium

Porównać działanie polecenia `ls` z opcją `-a` i bez tej opcji. Czy są jakieś różnice, a jeżeli tak to jakie?

Komenda `ls` służy do wypisania na terminal zawartości katalogu w którym aktualnie się znajdujemy. Domyślnie wypisywane są pliki i katalogi, które nie są plikami ukrytymi (zaczynającymi się od `.`). Flaga `-a` / `--all` powoduje że wypisane zostaną wszystkie pliki/katalogi, w tym ukryte, wraz z katalogami `.` oraz `..` reprezentującymi odpowiednio katalog w którym się aktualnie znajdujemy oraz katalog nadrzędny.

```
[baatochan@baatochan-virtualbox ~]$ ls
Desktop  Music          processes_16-25-52_2020-12-12.log  processes_16-44-01_2020-12-12.log  processes_16-53-02_2020-12-12.log  processes_16-58-01_2020-12-12.log  Videos
Documents new_file_crontab  processes_16-31-58_2020-12-12.log  processes_16-45-01_2020-12-12.log  processes_16-55-01_2020-12-12.log  processes_16-59-01_2020-12-12.log
Downloads Pictures      processes_16-40-17_2020-12-12.log  processes_16-46-01_2020-12-12.log  processes_16-56-01_2020-12-12.log  processes_16-57-01_2020-12-12.log  Public
Info.txt  processes_16-25-06_2020-12-12.log  processes_16-42-02_2020-12-12.log  processes_16-52-01_2020-12-12.log  processes_16-56-01_2020-12-12.log  Templates
[baatochan@baatochan-virtualbox ~]$ ls -a
.                .bashrc  Documents  Info.txt  new_file_crontab  processes_16-40-17_2020-12-12.log  processes_16-52-01_2020-12-12.log  processes_16-58-01_2020-12-12.log  .vboxclient-clipboard.pid  .Xclients
..               .cache   Downloads  .kde      Pictures          processes_16-42-02_2020-12-12.log  processes_16-53-02_2020-12-12.log  processes_16-59-01_2020-12-12.log  .vboxclient-draganddrop.pid  .xinitrc
.bash_history    .config  .face       .lesshst  processes_16-25-06_2020-12-12.log  processes_16-44-01_2020-12-12.log  processes_16-55-01_2020-12-12.log  Public                        .vboxclient-seamless.pid  .zshrc
.bash_logout     Desktop  .face.icon  .local    processes_16-25-52_2020-12-12.log  processes_16-45-01_2020-12-12.log  processes_16-56-01_2020-12-12.log  .ssh                          Videos
.bash_profile    .dir_colors  .gtkrcc-2.0  Music     processes_16-31-58_2020-12-12.log  processes_16-46-01_2020-12-12.log  processes_16-57-01_2020-12-12.log  Templates                    .Xauthority
```

Utworzyć w swoim katalogu domowym nowy katalog o takiej nazwie jak swoje nazwisko.

Tworzenie katalogu odbywa się poprzez komendę `mkdir <nazwa>`.

```
[baatochan@baatochan-virtualbox ~]$ mkdir rodziewicz
[baatochan@baatochan-virtualbox ~]$ cd rodziewicz/
[baatochan@baatochan-virtualbox rodziewicz]$ ls -aIf
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:20 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 11:20 ../
[baatochan@baatochan-virtualbox rodziewicz]$
```

Utworzyć nowy plik `p1.txt` i spróbować wykorzystać polecenie `chown` do zmiany jego właściciela. Co się dzieje i dlaczego?

Do wykonania zadania stworzony został nowy użytkownik `user1`.

Najprostszą metodą na utworzenie pliku jest użycie komendy `touch <nazwa>`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ touch p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 11:20 ../
-rw-r--r--  1 baatochan baatochan   0 Dec 18 11:40 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ chown user1 p1.txt
chown: changing ownership of 'p1.txt': Operation not permitted
[baatochan@baatochan-virtualbox rodziewicz]$
```

Przy próbie zmiany właściciela otrzymujemy komunikat o braku uprawnień. Dzieje się tak dlatego, że w systemie Linux, aby możliwa była zmiana właściciela muszą być spełnione dwa warunki:

- trzeba być właścicielem danego pliku,
- trzeba być "odbiorcą" uprawnień pliku.

Jak widać, powyższe warunki się wykluczają. Pierwszy z nich jest dość oczywisty i nie trzeba chyba tłumaczyć dlaczego trzeba być właścicielem pliku którego chcemy zmodyfikować.

Drugi natomiast wynika z faktu, że większość dystrybucji blokuje możliwość "oddawania" plików (można tylko przyjmować). Jest to spowodowane kilkoma powodami, gdzie dwa najważniejsze z nich to:

- Możliwość przekazania komuś pliku powoduje, że w systemie nie ma dowodu, że dany "obdarowany" użytkownik tego pliku nie stworzył. Mogłoby to być użyte by "podrzucić" komuś kompromitujące bądź nielegalne materiały.
- Możliwość przekazania komuś pliku powoduje problemy z działaniem quote'y na pliki danego użytkownika (limitu na zajęcie powierzchni dyskowej przez danego użytkownika). Możliwa byłaby sytuacja, gdy użytkownik tworzy plik w katalogu dostępnym tylko dla niego i przekazuje ten plik innej osobie. W efekcie plik zajmuje quote innego użytkownika, ale może być używany tylko przez pierwszego właściciela.

Oczywiście root posiada uprawnienia by zmieniać i przekazywać pliki każdemu.

Spróbować wykorzystać polecenie `chgrp` do zmiany grupy utworzonego w poprzednim punkcie pliku. Co się dzieje i dlaczego?

Do tego zadania wykorzystane zostaną dwie grupy - grupa `user1` stworzona wraz z użytkownikiem `user1` w poprzednim zadaniu oraz grupa `group1` stworzona dodatkowo. Grupa `user1` zawiera użytkownika `user1`, natomiast `group1` użytkownika `baatochan` i `user1`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan baatochan    0 Dec 18 11:40 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ chgrp user1 p1.txt
chgrp: changing group of 'p1.txt': Operation not permitted
[baatochan@baatochan-virtualbox rodziewicz]$ chgrp group1 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan group1      0 Dec 18 11:40 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Ja widąc na powyższym rzucie zmiana grupy na `user1` się nie powiodła z powodu braku uprawnień, natomiast zmiana na `group1` się powiodła. Stało się to z tego samego powodu o którym pisałem w zadaniu powyżej. Zamiana uprawnień jest możliwa tylko w dwóch przypadkach, dla grup będzie to:

- trzeba być właścicielem pliku,
- trzeba być "odbiorcą" uprawnień pliku, czyli trzeba być członkiem "obdarowanej" grupy.

Powody drugiego warunku pozostają takie jak w zadaniu wyżej. Tak samo jak dla `chown` root może zmieniać grupę bez żadnych przeciwwskazań.

Ustalić następujące uprawnienia do pliku prawa do p1.txt:

- właściciel ma możliwość tylko czytania pliku i jego wykonywania (dokładnie te dwa prawa)
- odebrać grupie wszelkie prawa za wyjątkiem możliwości wykonywania pliku
- pozostali nie mają żadnych uprawnień

Powyższe uprawnienia można zapisać w skrócie jako `510`. Kolejne cyfry oznaczają odpowiednio uprawnienia właściciela, grupy i innych. Cyfry to tak naprawdę liczby o wielkości 3 bitów, gdzie każdy bit odpowiada od lewej odczytowi, zapisowi i wykonywaniu. W dziesiętnym zapis to 4, odczyt 2, a wykonywanie 1. Poszczególne bity w zapisie dziesiętnym należy zsumować w naszym wypadku 4+0+1, 0+0+1, 0+0+0. Do zmiany uprawnień służy komenda `chmod`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan group1      0 Dec 18 11:40 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ chmod 510 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-r-x--x---  1 baatochan group1      0 Dec 18 11:40 p1.txt*
[baatochan@baatochan-virtualbox rodziewicz]$
```

Przywrócić właścicielowi pliku p1.txt komplet praw do pliku.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-r-x--x---  1 baatochan group1      0 Dec 18 11:40 p1.txt*
[baatochan@baatochan-virtualbox rodziewicz]$ chmod 644 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 11:40 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan group1      0 Dec 18 11:40 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Umieścić w pliku p1.txt następujący tekst: `plik1 <Imię Nazwisko> <numer indeksu>`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 8
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:16 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan group1      0 Dec 18 15:16 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ echo "plik1 Bartosz Rodziewicz 226105" > p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p1.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 12
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:16 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  1 baatochan group1     32 Dec 18 15:17 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Utworzyć dwa twarde dowiązania do pliku p1.txt pod nazwami p2.txt i p3.txt.

Do tworzenia linków służy komenda `ln`. Komenda `ln` bez żadnych parametrów tworzy link twardy.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 12
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:19 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--  1 baatochan group1     32 Dec 18 15:17 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ln p1.txt p2.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ln p1.txt p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 20
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:20 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p1.txt
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p2.txt
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Wyświetlić zawartość plików p1.txt i p3.txt. Czy jest ona taka sama? Dlaczego?

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 20
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:20 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p1.txt
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p2.txt
560610 -rw-r--r--  3 baatochan group1     32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p1.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$
```


Link twardy to bezpośrednia referencja do pliku poprzez jego inode. Każdy plik posiada inode zawierający swoje metadane oraz wskaźnik do fizycznego miejsca na dysku z danymi. Tworząc link symboliczny tworzona jest druga referencja do tego samego inode. Przeniesienie lub skasowanie jednej referencji nie ma wpływu na pozostałe. Twarde linki można tworzyć tylko do plików, nie można do katalogów.

Na powyższych zrzutach widać, że każdy z plików ma ten sam inode (pierwsza kolumna `11 -i`).

Usunąć plik p1.txt i ponownie wyświetlić zawartość pliku p3.txt.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 20
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:20 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p1.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p2.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ rm p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 16
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:30 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  2 baatochan group1      32 Dec 18 15:17 p2.txt
-rw-r--r--  2 baatochan group1      32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$
```

Skasowanie jednej referencji nie ma wpływu na inne. Aby zdeakować plik na dysku konieczne jest usunięcie wszystkich referencji.

Zmienić zawartość pliku p2.txt na: `plik2 <Nazwisko Imię> <numer indeksu>`. W kolejnym kroku wyświetlić zawartość plików p2.txt i p3.txt. Co się stało i dlaczego?

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 20
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:20 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p1.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p2.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ rm p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll
total 16
drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:30 ./
drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
-rw-r--r--  2 baatochan group1      32 Dec 18 15:17 p2.txt
-rw-r--r--  2 baatochan group1      32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 16
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:30 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:17 p2.txt
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:17 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik1 Bartosz Rodziewicz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ echo "plik2 Rodziewicz Bartosz 226105" > p2.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 16
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:30 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:33 p2.txt
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:33 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Modyfikacja zawartości pliku przez jedną referencję powoduje modyfikację pliku widzianego przez każdą referencję.

Utworzyć twarde dowiązanie o nazwie p1.txt wskazujące na zawartość pliku p2.txt, a następnie usunąć pliki p2.txt i p3.txt. Jaka jest zawartość p1.txt i dlaczego?

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 16
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:30 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:33 p2.txt
560610 -rw-r--r--   2 baatochan group1      32 Dec 18 15:33 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ln p2.txt p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 20
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:38 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:33 p1.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:33 p2.txt
560610 -rw-r--r--   3 baatochan group1      32 Dec 18 15:33 p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p1.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p3.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ rm p2.txt p3.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 12
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 15:38 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   1 baatochan group1      32 Dec 18 15:33 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p1.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$
```

Skasowanie jednej referencji nie ma wpływu na inne. Aby zdealokować plik na dysku konieczne jest usunięcie wszystkich referencji. Nie ma znaczenia czy referencja jest tworzona do oryginalnego pliku (oryginalnej referencji) czy do którejś innej. Każda referencja (twardy link) jest równa innej referencji.

Utworzyć link symboliczny do pliku p1.txt pod nazwą p2.txt.

Symboliczne linki również można utworzyć komendą `ln`, należy użyć flagi `-s`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 12
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 21:59 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   1 baatochan group1      32 Dec 18 15:33 p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ln -s p1.txt p2.txt
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 12
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 21:59 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   1 baatochan group1      32 Dec 18 15:33 p1.txt
560616 lrwxrwxrwx   1 baatochan baatochan    6 Dec 18 21:59 p2.txt -> p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Wyświetlić zawartość pliku p2.txt Czy jest taka sama jak pliku p1.txt?

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll -i
total 12
560606 drwxr-xr-x  2 baatochan baatochan 4096 Dec 18 21:59 ./
418034 drwx----- 16 baatochan baatochan 4096 Dec 18 13:34 ../
560610 -rw-r--r--   1 baatochan group1      32 Dec 18 15:33 p1.txt
560616 lrwxrwxrwx   1 baatochan baatochan    6 Dec 18 21:59 p2.txt -> p1.txt
[baatochan@baatochan-virtualbox rodziewicz]$ cat p1.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$ cat p2.txt
plik2 Rodziewicz Bartosz 226105
[baatochan@baatochan-virtualbox rodziewicz]$
```

Na powyższym zrzucie widzimy, że zawartość obu plików jest identyczna. Link symboliczny to osobny plik (typu link) będący wskaźnikiem do ścieżki oryginalnego pliku. Oznacza to, że podmiana oryginalnego pliku zmieni jego zawartość. Korzystając z symbolicznego linku korzystamy tak jakby ze skrótu do oryginalnego pliku. Link symboliczny może być używany również do katalogów, jak i do plików/katalogów znajdujących się na innym woluminie.

Próba wyświetlenia zawartości kończy się informacją, że taki plik nie istnieje. Jest to tak zwany dangling link, czyli sytuacja w której link symboliczny wskazuje na plik, który nie istnieje. Stworzenie nowego pliku o nazwie na którą wskazuje link spowodowałoby, że link zacząłby działać z powrotem. Link symboliczny zawiera jedynie wskaźnik na plik oryginalny.

Wykorzystując polecenie `mount` oraz `fdisk` sprawdzić jakie systemy plików są aktualnie zamontowane. Porównać informacje dostarczane przez oba polecenia.

[illegible]

Zamontować pendrive podłączony do portu USB. Sprawdzić, czy operacja się powiodła.

Po podłączeniu pendrive'a i przekazaniu go do VM wyświetliło się stosowne powiadomienie z GUI, które umożliwiało automatyczne montowanie nowego urządzenia. Montowanie z poziomu terminala jest możliwe przy użyciu komendy `mount <urządzenie> <miejsce>`. Katalog w którym chcemy zamontować musi już istnieć. Montowanie w `/mnt/` wymaga uprawnień roota. Komenda `mount` od jakiegoś czasu nie wymaga podawania typu systemu plików. Aby zamontować dane urządzenie w trybie read-only konieczne jest użycie flagi `-r`, domyślnie następuje mount read-write.

```
[baatochan@baatochan-virtualbox rodziewicz]$ ll /dev/sd*
brw-rw---- 1 root disk 8,  0 Dec 18 10:43 /dev/sda
brw-rw---- 1 root disk 8,  1 Dec 18 10:43 /dev/sda1
brw-rw---- 1 root disk 8, 16 Dec 18 22:33 /dev/sdb
brw-rw---- 1 root disk 8, 17 Dec 18 22:33 /dev/sdb1
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt
total 12
drwxr-xr-x  2 root root 4096 Sep 26 14:06 ./
drwxr-xr-x 17 root root 4096 Dec  7 13:22 ../
[baatochan@baatochan-virtualbox rodziewicz]$ sudo mount /dev/sdb1 /mnt/usbdrive
mount: /mnt/usbdrive: mount point does not exist.
[baatochan@baatochan-virtualbox rodziewicz]$ sudo mkdir /mnt/usbdrive
[baatochan@baatochan-virtualbox rodziewicz]$ sudo mount /dev/sdb1 /mnt/usbdrive
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt
total 16
drwxr-xr-x  3 root root 4096 Dec 18 22:38 ./
drwxr-xr-x 17 root root 4096 Dec  7 13:22 ../
drwxr-xr-x  6 root root 4096 Jan  1  1970 usbdrive/
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt/usbdrive/
total 2336
drwxr-xr-x  6 root root  4096 Jan  1  1970  ./
drwxr-xr-x  3 root root  4096 Dec 18 22:38  ../
-rwxr-xr-x  1 root root  1558 Dec  3  2018  [redacted]
-rwxr-xr-x  1 root root    56 Sep 15  2016  [redacted]
-rwxr-xr-x  1 root root 355094 Feb 26  2018  [redacted]
drwxr-xr-x  3 root root  4096 Feb 26  2018  [redacted]
-rwxr-xr-x  1 root root 1872472 Oct 27  2016  [redacted]
drwxr-xr-x  5 root root  4096 Oct 23  2017  [redacted]
drwxr-xr-x  9 root root  4096 Nov 13  2017  [redacted]
-rwxr-xr-x  1 root root 121716 Nov 29  2018  [redacted]
drwxr-xr-x  2 root root  4096 Nov  6  2015  [redacted]
-rwxr-xr-x  1 root root   272 Oct 25  2017  [redacted]
[baatochan@baatochan-virtualbox rodziewicz]$
```

Z uwagi na podłączenie prywatnego pendrive'a pozwoliłem sobie zabluirować jego zawartość.

Zamontować CD-rom w sposób "pedantyczny", tzn. jawnie podając odpowiedni system plików.

```
[baatochan@baatochan-virtualbox rodziewicz]$ sudo mkdir /mnt/cd
[baatochan@baatochan-virtualbox rodziewicz]$ sudo mount -rt iso9660 /dev/sr0 /mnt/cd
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt
total 18
drwxr-xr-x  4 root root 4096 Dec 18 22:55 ./
drwxr-xr-x 17 root root 4096 Dec  7 13:22 ../
dr-xr-xr-x  1 root root 2048 Aug  8 04:33 cd/
drwxr-xr-x  6 root root 4096 Jan  1  1970 usbdrive/
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt/cd
total 416
dr-xr-xr-x  1 root root  2048 Aug  8 04:33  ./
drwxr-xr-x  4 root root  4096 Dec 18 22:55  ../
dr-xr-xr-x  1 root root  2048 Aug  8 04:33  boot/
-r--r--r--  1 root root  2048 Aug  8 04:08  boot.catalog
dr-xr-xr-x  1 root root  2048 Aug  8 04:33  .disk/
dr-xr-xr-x  1 root root  2048 Aug  8 04:33  EFI/
-r--r--r--  1 root root 327680 Aug  8 04:33  efi.img
dr-xr-xr-x  1 root root  2048 Aug  8 04:32  live/
-r--r--r--  1 root root  81668 Aug  8 04:33  sha256sum.txt
[baatochan@baatochan-virtualbox rodziewicz]$
```

Montowanie płyty CD odbywa się w podobny sposób co powyżej z wyjątkiem flagi `-t`, która służy do jawnego specyfikowania typu plików urządzenia, które montujemy.

Co oznaczają poszczególne pola `fstab` z poniższego przykładu:

```
/dev/hda1 /mnt/dysk ntfs rw,uid=1000,codepage=852,iocharset=iso8859-2,auto 0 2
```

Fragment wpisu	Znaczenie
/dev/hda1	Nazwa urządzenia/partycji
/mnt/dysk	Miejsce montowania
ntfs	Typ systemu plików
rw	Montowanie w trybie read-write (odczyt-zapis)
uid=1000	Montowanie z ustawieniem użytkownika o uid 1000 jako właściciela wszystkich plików na danym wolumenie. Używane głównie do systemów ntfs i fat i innych systemów nie definiujących właściciela każdego pliku w sposób linuxowy.
codepage=852,iocharset=iso8859-2	Definicja kodowania znaków. <code>codepage</code> jest używany w krótkich nazwach, <code>iocharset</code> w dłuższych.
auto	Automatyczne montowanie przy starcie systemu
0	Wykorzystywane przez komendę <code>dump</code> do backupów. 0 oznacza pominięcie
2	Wykorzystywane przez komendę <code>fsck</code> by określić czy dany system plików ma być sprawdzony przy uruchamianiu i w jakiej kolejności. 0 oznacza brak sprawdzenia, 1 jest zarezerwowany dla partycji root, 2 dla wszystkich innych.

Odmontować systemy plików zamontowane w poprzednich ćwiczeniach oraz usunąć ewentualne wpisy w `fstab`.

```
[baatochan@baatochan-virtualbox rodziewicz]$ sudo umount /mnt/usbdrive/
[sudo] password for baatochan:
[baatochan@baatochan-virtualbox rodziewicz]$ sudo umount /mnt/cd/
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt/
total 20
drwxr-xr-x  4 root root 4096 Dec 18 22:55 ./
drwxr-xr-x 17 root root 4096 Dec  7 13:22 ../
drwxr-xr-x  2 root root 4096 Dec 18 22:55 cd/
drwxr-xr-x  2 root root 4096 Dec 18 22:38 usbdrive/
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt/cd
total 8
drwxr-xr-x 2 root root 4096 Dec 18 22:55 ./
drwxr-xr-x 4 root root 4096 Dec 18 22:55 ../
[baatochan@baatochan-virtualbox rodziewicz]$ ll /mnt/usbdrive/
total 8
drwxr-xr-x 2 root root 4096 Dec 18 22:38 ./
drwxr-xr-x 4 root root 4096 Dec 18 22:55 ../
[baatochan@baatochan-virtualbox rodziewicz]$
```

Powyższy zrzut pokazuje odmontowanie obu urządzeń. W pliku `/etc/fstab` nie były dokonywane żadne modyfikacje.