

Linux, część druga

Na pierwszych zajęciach poznaliśmy totalne podstawy posługiwania się wierszem poleceń, przynosiliśmy pliki, ogarnęliśmy ścieżki, kopiowanie i różne takie. Tych, którzy zapomnieli odsyłam do skryptu z laboratorium 1.

Na dzisiejszych zajęciach zajmiemy się bardziej zaawansowanymi metodami pracy z terminalem, nadamy katalogom i plikom uprawnienia, zastosujemy filtry i popracujemy trochę z tekstem.

Wildcards, czyli znaki specjalne

Czasem może zdarzyć się sytuacja, w której potrzebujemy wykonać operację na wielu plikach, które mają jakiś element wspólny. Takimi plikami mogą być na przykład logi aplikacji, przepisy na lasagne, czy pliki o określonym rozszerzeniu. Tutaj z pomocą przychodzą nam tzw. wildcards. W uproszczeniu zastępują one jeden lub wiele znaków. Na przykład:

```
ls *.txt
```

Wyszuka nam wszystkie pliki txt w katalogu, w którym wykonamy polecenie.

```
michal@michal-VirtualBox:~/Dokumenty$ ls *.txt
jajco.txt  obiad.txt  plan_nuklearnej_zaglady.txt
michal@michal-VirtualBox:~/Dokumenty$
```

Podstawowymi znakami specjalnymi są:

- * - dowolna ilość dowolnych znaków
- ? - jeden dowolny znak
- [] - umieszczony wewnątrz zakres znaków

Dla przykładu, aby znaleźć wszystkie Lasagne w naszym katalogu z przepisami użyjemy polecenia

```
ls Lasagne*
```

```
michal@michal-VirtualBox:~/Dokumenty/Obiady$ ls Lasagne*
Lasagne_babuni.pdf  Lasagne_bolognese.txt  Lasagne_ze_szpinakiem.txt
michal@michal-VirtualBox:~/Dokumenty/Obiady$
```

Pamiętając jednak, że Linux jest case-sensitive, wiemy, że mogą być tu jakieś pliki z lasagne, których nazwa nie zawiera ciągu znaków Lasagne, tylko lasagne. W tym miejscu przyda nam się znak zapytania.

```
ls ?asagne*
```

W ten sposób określamy, że szukamy tylko i wyłącznie plików i katalogów, które zaczynają się od jakiegoś, dowolnego znaku, potem następuje ciąg znaków „asagne”, a po nim może być cokolwiek.

```
michal@michal-VirtualBox:~/Dokumenty/Obiady$ ls -la ?asagne*
-rw-r--r-- 1 michal michal 0  lis 14 02:09 Lasagne_babuni.pdf
-rw-r--r-- 1 michal michal 0  lis 14 02:09 Lasagne_bolognese.txt
-rw-r--r-- 1 michal michal 0  lis 14 02:13 lasagne_jest_super.cpp
-rw-r--r-- 1 michal michal 0  lis 14 02:09 Lasagne_ze_szpinakiem.txt
michal@michal-VirtualBox:~/Dokumenty/Obiady$
```

Pozostaje nam zastosowanie ostatniego wildcarda, nawiasów kwadratowych []. W tym nawiasie możemy określić znaki, bądź ich zakres.

```
ls [jkm]*
```

wyszuka nam wszystkie pliki i katalogi, które zaczynają się od litery j, k lub m;

```
ls *[0-5]*
```

wyszuka nam pliki i katalogi, które zawierają będą cyfrę od 0 do 5;

```
ls [^h-n]*
```

wyszuka nam pliki i katalogi, które NIE ZACZYNAJĄ się od litery w zakresie od h do n. Za odwrócenie takiego ciągu odpowiada znak ^

```
Michal@Michal-VirtualBox:~/Dokumenty/Log$ ls
log1 log10 log120 log2 log3 log4 log5 log6 log7 log8 log9 logo.bmp
Michal@Michal-VirtualBox:~/Dokumenty/Log$ ls log[0-4]
log1 log2 log3 log4
Michal@Michal-VirtualBox:~/Dokumenty/Log$
```

Ważna uwaga: wildcardy są częścią powłoki, nie poleceń, bądź programu. Możemy ich w ten sposób używać w terminalu dla dowolnego polecenia. Dla przykładu życiowego zastosowania, przeniesiemy teraz wszystkie przepisy na lasagne z katalogu Obiady do katalogu Przepisy

```
mv Obiady/?asagne* Przepisy
```

```
Michal@Michal-VirtualBox:~/Dokumenty/Przepisy$ ls
history.txt      Lasagne_babuni.pdf      Obiady
Hotdog.txt       Lasagne_bolognese.txt  pizza
jajco.txt        lasagne_jest_super.cpp  'Quiche Lorraine.txt'
Jajecznicza.txt Lasagne_ze_szpiniakiem.txt
Michal@Michal-VirtualBox:~/Dokumenty/Przepisy$
```

W identyczny sposób możemy wyszukać dla przykładu wszystkie pliki tekstowe w katalogach podrzędnych od tego, w którym się znajdujemy

```
Michal@Michal-VirtualBox:~$ ls */*.txt
Dokumenty/jajco.txt      PJATK/jajco.txt
Dokumenty/obiad.txt     'PJATK/Quiche Lorraine.txt'
Dokumenty/plan_nuklearnej_zaglady.txt
Michal@Michal-VirtualBox:~$
```

Zadanie na rozgrzewkę

Utwórz dwa katalogi o dowolnych nazwach. Dla potrzeb tego ćwiczenia nazwę je Katalog1 i Katalog 2. W Katalogu1 wykonaj polecenie

```
touch list.txt lost last.txt tost.txt ticket.txt post.txt
```

Twoim zadaniem jest przekopiować (cp) pliki list.txt, lost oraz tost.txt do drugiego katalogu używając tylko jednego polecenia.

Uprawnienia

Z różnymi poziomami uprawnień spotkać się możemy na co dzień korzystając z komputera, chociażby na sprzęcie uczelnianym, gdzie nikt nie ma praw do niczego. W systemie Linux rozróżniamy trzy rodzaje uprawnień, które możemy nadać trzem grupom użytkowników. Polecenie

```
ls -la
```

wypisze nam listę plików. Pierwsze 10 znaków w wierszu oznaczają właśnie uprawnienia. Teraz je odczytamy.

```
Michal@michal-VirtualBox:~/Dokumenty$ ls -la
razem 32
drwxr-xr-x  7 michal michal 4096 lis 14 02:26 .
drwxr-xr-x 23 michal michal 4096 lis 14 02:49 ..
-rw-r--r--  1 michal michal   0 paź  3 13:20 jajco.txt
drwxr-xr-x  2 michal michal 4096 paź  4 16:31 Kolacje
drwxr-xr-x  2 michal michal 4096 lis 14 02:27 Log
-rw-r--r--  1 michal michal   0 paź  3 03:07 najlepsze_browary_2019.doc
-rw-r--r--  1 michal michal   9 paź  3 12:53 obiad.txt
drwxr-xr-x  2 michal michal 4096 lis 14 02:41 Obiady
-rw-r--r--  1 michal michal   0 paź  3 03:07 plan_nuklearnej_zaglady.txt
drwxr-xr-x  3 michal michal 4096 lis 14 02:41 Przepisy
drwxr-xr-x  2 michal michal 4096 paź  4 16:31 Sniadania
```

Pierwszy znak sygnalizuje, czy mamy do czynienia z katalogiem, czy plikiem. d oznacza katalog (directory), '-' oznacza plik. Dalej następuje ciąg 9 znaków, rozkodujemy go teraz.

1. **r**ead – uprawnienia do odczytywania pliku
2. **w**rite – uprawnienia do zmiany zawartości pliku
3. **x**ecute – uprawnienia do wykonania pliku

Dla każdego pliku określamy trzy grupy użytkowników:

1. **u**ser – właściciel, zwykle osoba, która utworzyła plik, choć niekoniecznie
2. **g**roup – grupa
3. **o**ther – pozostali

znaki odczytujemy standardowo, od lewej strony. Pierwszy znak określa nam czy mamy do czynienia z katalogiem czy plikiem, kolejne 3 znaki to uprawnienia, które posiada właściciel, kolejne 3 to uprawnienia grupy, a 3 ostatnie to uprawnienia pozostałych.

Nadawanie i odbieranie uprawnień

Do zmiany uprawnień pliku/katalogu służy polecenie

```
chmod [uprawnienia] [ścieżka]
```

argumenty uprawnień składają się natomiast z 3 kolejnych elementów:

1. Komu zmieniamy uprawnienia? [ugoa] (user, group, others, all)
2. Nadajemy, czy zabieramy uprawnienia: (+) nadaje, (-) odbiera
3. Jakie uprawnienia zmieniamy? [rwx]

Tak więc, aby nadać uprawnienia do zapisu pliku obiady.txt pozostałym użytkownikom użyjemy polecenia

```
chmod o+w obiad.txt
```

Po wykonaniu polecenia możemy poleceniem ls sprawdzić, czy faktycznie uprawnienia zostały zmienione

Filtrowanie

Czym w ogóle są filtry? W linii poleceń linuxa możemy określić je jako programy, które pobierają tekst oraz transformują go w zadany sposób. Podstawowymi filtrami są:

```
head [-liczba linii do wyświetlenia] [ścieżka]
```

Wyświetli nam on pierwsze x linii w pliku. Domyślna ilość linii do wyświetlenia to 10;

```
tail [-liczba linii do wyświetlenia] [ścieżka]
```

To zupełne przeciwieństwo polecenia head. Wyświetla ostatnich x linii;

```
sort [-opcje] [ścieżka]
```

Domyślnie posortuje nam zawartość pliku alfabetycznie, ale zachęcam do sprawdzenia manuala, opcji jest multum;

```
nl [-opcje] [ścieżka]
```

ponumeruje linie;

Powyższe to tylko przykłady, ponieważ opcji sortowania jest naprawdę sporo, jak chociażby *sed*, *cut*, *uniq* czy *tac*. Odsyłam do manuala, aby poznać ich dokładne zastosowanie, przyda się do zadania.

Zadanie 2

Utwórz plik z listą zakupów w formacie

```
[Sklep] [produkt] [ilość]
```

Czyli dla przykładu:

```
Lidl ogórki 7  
Lidl jajka 10  
Auchan Coca-Cola 12  
E.Leclerc Browar 128
```

Niech tych pozycji będzie co najmniej 15. Następnie spróbuj wyświetlić tylko zakupy, które zrobić trzeba w Lidlu. Jeszcze raz: odsyłam do manuala!