



Linux

Polecenia

Opracował: Andrzej Nowak

Problem nadpisywania plików. Zmienna noclobber i noglob. Filtry i metaznaki.

Utwórz katalog **lab_5** i przejdź do niego.

```
$ mkdir lab_5 ; cd lab_5
```

Problem nadpisywania plików

Sprawdź czy jesteś w bash'u

```
$ w
```

```
$ w > ktoto
```

```
$ cat ktoto
```

```
$ date > ktoto
```

```
$ cat ktoto
```

PROBLEM?

```
$ ps
```

```
$ tcsh
```

```
$ ps
```

```
$ set -/wpisz nazwę pliku/ noclobber
```

```
$ ls > ktoto
```

```
ktoto: File exists
```

```
$ cat ktoto
```

```
$ ls >! ktoto
```

Zmienna **noclobber** służy do zabezpieczenia własnych plików przed nadpisaniem ich przez inny program lub przekierowaniem. Zachowuje oryginalny plik w nienaruszonym stanie.

```
$ cat ktoto
```

```
$ ps
```

```
$ exit
```

```
$ ps
```

```
$ date > ktoto
```

```
$ cat ktoto
```

Polecenie tee

```
$ date
```

```
$ date | wc
```

```
$ date; who
```

```
$ who; date | wc
```

```
$ (who; date) | wc
```

```
$ (date; who) | tee listop
```

tee - przesyła swoje dane wejściowe na standardowe wyjście i do pliku listop

```
$ cat listop
```

Filtry

Filtr jest poleceniem, które przetwarza strumień danych wejściowych i produkuje strumień danych wyjściowych.

```
$ date | tee data.out
```

```
$ cat data.out
```

```
$ date | tee data.out | wc
```

Można używać wielu potoków w poleceniu

Utwórz plik imiona:

```
kasia  
janusz  
dominik  
ola
```

Metaznaki

Używając polecenia **cat** lub **touch** utwórz następujące pliki:

```
abc, abc12, abc2, file1.bak, file2.bak nonsense, nothing, one,  
abc1, abc123, file1, file2, none, noone, nowhere, masza
```

a) metaznak *

```
$ ls  
$ ls *  
$ ls *.bak  
$ echo a*
```

metaznak ***** reprezentuje ciąg znaków składających się z 0 lub więcej dowolnych znaków w nazwie pliku.

b) metaznak ?

```
$ ls  
$ ls a?c?  
$ ls ??  
$ echo abc???  
$ echo ??
```

metaznak **?** reprezentuje pojedynczy znak w nazwie pliku.

c) metaznaki [abc] [a-z] [!a-z]

```
$ ls  
$ ls abc[123]  
$ ls abc[1-3]  
$ ls [a-z][a-z][a-z]
```

```
$ ls [!f-z]???
```

```
$ ls abc12[23]
```

```
$ set -o noglob
```

```
$ ls file[12].bak
```

```
$ set +o noglob
```

```
$ ls file[12].bak
```

Ustawienie **noglob** powoduje wyłączenie możliwości używania metaznaków w powłoce użytkownika.

[abc] - reprezentuje jeden znak ze zbioru {a,b,c}

[a-z] - reprezentuje jeden znak z zakresu od a do z

[!a-z] - reprezentuje jeden znak nie będący w zakresie od a do z

**d) metaznak **

```
$ ls
```

```
$ echo Jak sie masz?   !!!(masza)
```

```
$ echo Jak sie masz\?
```

```
$ echo Czy ta linia\ [Enter]
```

```
> kiedykolwiek sie zakonczy\?
```

Jeżeli chcemy użyć dany metaznak w sensie dosłownym to \ może być użyty, aby dany metaznak nie był interpretowany.

e) niektóre metaznaki poznaliśmy już wcześniej: ; () | > >> &

```
$ (sleep 10; who)& date | wc > nowa_data
```

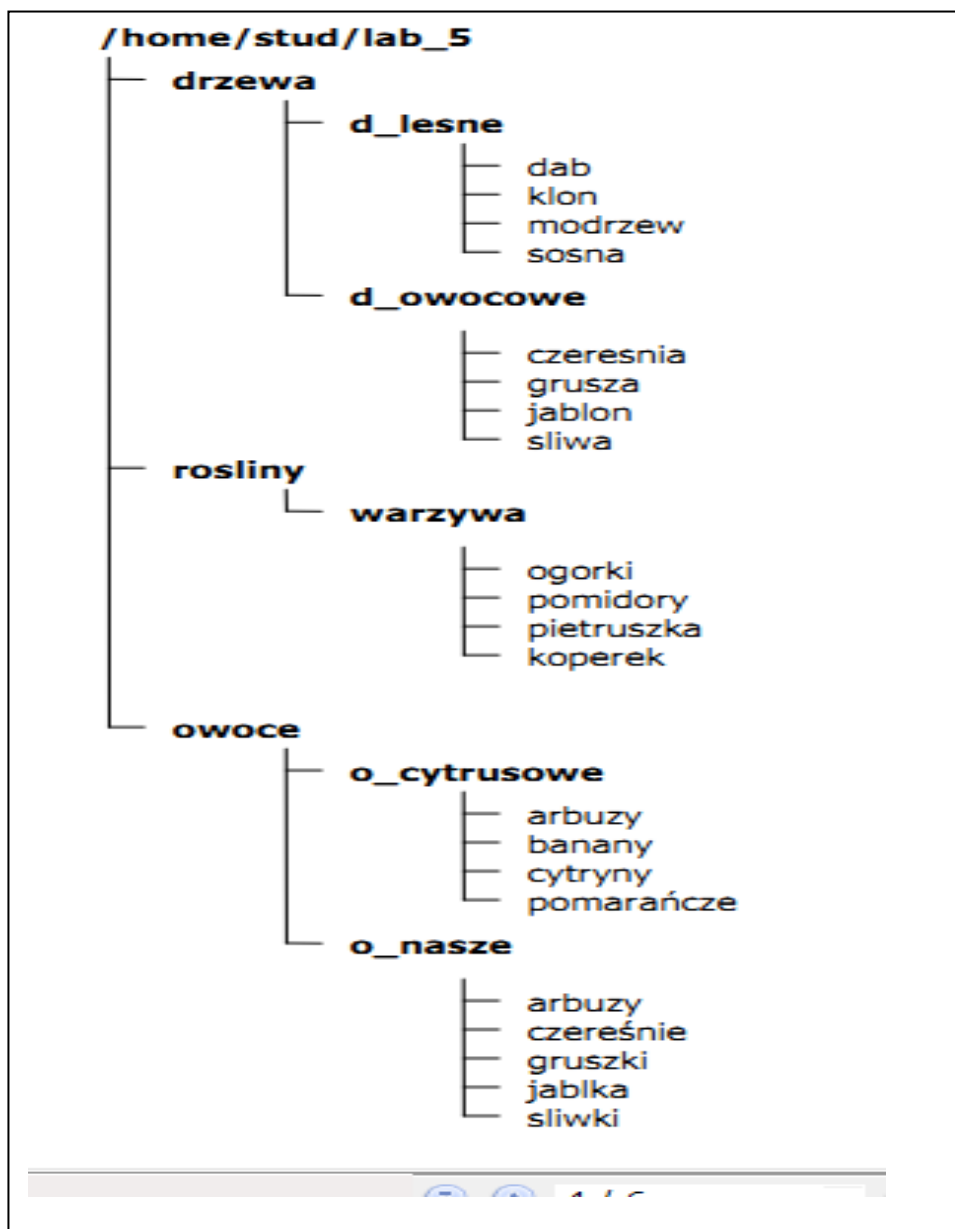
Polecenie find

Polecenie to jest niezwykle użyteczne dla odszukiwania poszczególnych grup plików:

```
find ścieżka(i) warunek(i)
```

Polecenie to powoduje rekurencyjne przeszukiwanie drzewa katalogów, począwszy od każdej ścieżki i powoduje odnalezienie plików, które spełniają zadane warunki.

Używając poleceń **mkdir** i **touch** utwórz następujące drzewo katalogów i plików (katalogi są pogrubione):



Używając edytora zmień zawartość plików:

jablka

```
ja lubie
mama nie
tata tak
```

gruszki

```
ja nie lubie
mama lubi
tata lubi
```

Przejdź do katalogu **lab_5**

- warunek **-print**

Wyświetla na ekranie odnalezione pliki i katalogi używając pełnych ścieżek w charakterze ich nazw.

```
$ find $HOME -print
```

```
$ find $HOME -print | more
```

- warunek **-name wzorzec**

Odnajduje pliki, których nazwy spełniają wzorzec. Metaznaki mogą być tu używane - trzeba tu pamiętać o zasadach cytowania.

```
$ find . -name '*e' -print
```

```
$ find ./owoce -name arbuzy -print
```

```
$ find . \! -name '[a-r]*' -print
```

```
$ find . \( -name arbuzy -o -name 's*' \) -print
```

- warunek **-size n**

Odnajduje pliki zawierające n bloków

```
$ find ./drzewa -size 0 -print
```

- warunek **-type c**

Odnajduje pliki typu **c**, gdzie **c** może przyjmować w szczególności takie wartości:

o **f** - zwykły plik

o **d** - katalog

```
$ find . -type d -print
```

```
$ find . -type f -print
```

```
$ find / -type d -name 'linux*' -print
```

- warunek **-newer plik**

Odnajduje pliki, które były modyfikowane później niż dany **plik**

Przejdź do katalogu **d_owocowe**

```
$ ls -l
```

```
$ find . -newer sliwa -print
```

```
$ touch sliwa
```

```
$ ls -l
```

```
$ find . -newer jablon -print
```

- polecenie **-exec polecenie { } \;**

Wykonuje UNIX-owe **polecenie** na każdym pliku odnalezionym przez **find**; nawiasy będą zamienione na odnalezione pliki.

Powrót do katalogu **lab_5**.

```
$ find . -type f -exec grep 'mama' { } \; -print
$ find . -size 0 -print
$ find . -size 0 -ok rm { } \;
$ find . -size 0 -print
$ find . \! -size 0 -ok rm { } \;
```

Sprawdź czy zostały jakieś pliki w **lab_5**.

Ćwiczenie do samodzielnego wykonania

Utwórz katalog **feb20** i przejdź do niego, a następnie używając polecenia **touch** utwórz w nim następujące pliki:

ab abc a1 a2 a3 all ba ba.1 ba.2 filex filey AbC ABC ABC2 abc

Napisz polecenia które wyświetlą:

- a) wszystkie pliki zaczynające się od a,
- b) wszystkie pliki kończące się co najmniej jedną cyfrą,
- c) wszystkie pliki zaczynające się z a lub A,
- d) wszystkie pliki kończące się . oraz następującą po niej cyfrą,
- e) wszystkie pliki zawierające tylko dwa znaki,
- f) pliki, których nazwy składają się z trzech znaków – dużych liter,
- g) pliki kończące się na 10, 11 lub 12,
- h) pliki kończące się na x lub y,
- i) pliki kończące się cyfrą, dużą literą lub małą literą,
- j) pliki nie zaczynające się na b lub B,
- k) usuń pliki, których nazwy składają się z dwóch znaków i zaczynają się na a lub A.

Polecenia zapisz do pliku o nazwie polecenia
