

Wstęp do informatyki

Shell (cz.3)

Cudzysłowy, apostrofy, etc

Podobnie do `$(...)` działa `$((...))`.

`$((...))` – wykonuje obliczenie wyrażenia arytmetycznego i podstawia wynik obliczenia z `stdout`.

```
> echo $ ( ( 2+2 ) )  
4
```

Zagadka :

```
> echo $ ( ( `seq -s '*' 10` ) )
```

Stara wersja to `$(...)` ale w przyszłych wersjach basha już takiej nie będzie.

Grep

grep [opcje] pliki

Wypisz linie z plików/pliku pasujące do wzorca

```
> grep Usage *.sh
> grep '^echo' *.sh # na początku linii
> grep 'do$' *.sh # na końcu linii
> grep [Ss]ortuj *.txt
> grep '[A-Z]....' *.txt # 5-literowe frazy
```

Grep

- > `grep "(Pawel\|Gorecki)" /etc/passwd`
- > `grep "Pawel" - # czytaj z std in`

- > `grep -l Wzorzec pliki`
lista pasujących plików

- > `grep -v Wzorzec pliki`
pokaż niepasujące linie

Użyteczne przykłady

Przekierowanie do cat. Zmienne są rozwijane (tutaj \$0):

```
function Usage()  
{  
  
cat << EOF  
$0 [-h] [-c]  
-h - help  
-c - copy  
EOF  
  
}
```

Użyteczne przykłady

Przekierowanie do cat, cd.

```
cat << EOF > wazne.txt  
Hello world!  
Zajęcia z WI są świetne!  
EOF
```

Użyteczne przykłady

sort

Sortuj linie w pliku.

Sortuj plik.txt, białe spacje oddzielają kolejne pola.

```
> sort plik.txt
```

Sortuj numerycznie (-n), klucz to 1 pole.

```
> sort -n -k1 plik.txt
```

Klucz to 2 i 4 pole, separator to dwukropek

```
> sort -t: -k2,3 plik.txt
```

Sortuj z pominięciem duplikatów

```
> cat plik.txt | sort -u
```

Użyteczne przykłady

uniq

Usuń sąsiadujące powtórzenia linii

```
> uniq -u plik.txt
```

Często stosowane z sort – policz wystąpienia linii

```
> sort plik.txt | uniq -c
```

Praca domowa:

```
> sort -u
```


Użyteczne przykłady

Jak czytać z pliku?

Odp. polecenie wbudowane `read`, które czyta jedną linię z wejścia i zapisuje do zmiennych. Separatory to białe znaki.

Użyteczne przykłady

find

Znajdź pliki.

```
> find -name '*.txt'
```

```
> find -name '*.bak' -exec rm {} \;
```

```
> find ! -name '*.txt'
```

```
> find -size +1000k
```

```
> find -size +1000k -exec ls -l {} \;
```

Użyteczne przykłady

cut

Wytnij kolumny (domyślny separator TAB)

Pokaż kolumny 1,3,4 z polecenia ls

```
> ls -l | cut -f1,3-4
```

Pokaż kolumnę 2, separator :

```
> cut -f2 -d" :"
```