

Podstawowy warsztat informatyka

PWI

Instytut Informatyki Uniwersytetu Wrocławskiego

Wykład 5

częściowo na podstawie slajdów Jakuba Michaliszyna

Praca zdalna: ssh, scp - demo

Wyszukiwanie danych

grep kopytko

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuere non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.

Wyszukiwanie danych

grep kopytko

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuere non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.

ac massa. Quisque non **makopytko** vel orci efficitur posuere

Wyrażenia regularne

<http://www.tldp.org/LDP/abs/html/x17129.html>

Wyrażenia regularne

- Każdy znak dopasowuje się do siebie (oprócz specjalnych).

Wyrażenia regularne

- Każdy znak dopasowuje się do siebie (oprócz specjalnych).
- Kropka – dowolny znak (czasem z wyjątkiem znaku nowego wiersza)

Wyrażenia regularne

- Każdy znak dopasowuje się do siebie (oprócz specjalnych).
- Kropka – dowolny znak (czasem z wyjątkiem znaku nowego wiersza)
- `Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuer non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.`

Wyrażenia regularne

- Każdy znak dopasowuje się do siebie (oprócz specjalnych).
- Kropka – dowolny znak (czasem z wyjątkiem znaku nowego wiersza)
- `Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuer non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.`
- `grep ni.`

Wyrażenia regularne

- Każdy znak dopasowuje się do siebie (oprócz specjalnych).
- Kropka – dowolny znak (czasem z wyjątkiem znaku nowego wiersza)
- `Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuer non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.`
- `grep ni.`
- `Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices at eros. Proin dictum placerat nibh, a fringilla ipsum`

Wyrażenia regularne

- Znaki specjalne poprzedzone odwrotnym ukośnikiem `\` oznaczają same siebie, np. `\.` oznacza kropkę.

Wyrażenia regularne

- Znaki specjalne poprzedzone odwrotnym ukośnikiem \ oznaczają same siebie, np. \. oznacza kropkę.
- ```
grep i\.,
grep i\\..,
grep "i\."
```

## Wyrażenia regularne

- Znaki specjalne poprzedzone odwrotnym ukośnikiem \ oznaczają same siebie, np. \. oznacza kropkę.
- ```
grep i\.,  
grep i\\..,  
grep "i\."
```
- Gwiazdka po symbolu (nawiasie, pojedynczym znaku) nazywana jest domknięciem Kleene'a i oznacza zero lub więcej wystąpień poprzedzającego wyrażenia.

Wyrażenia regularne

- Znaki specjalne poprzedzone odwrotnym ukośnikiem \ oznaczają same siebie, np. \. oznacza kropkę.
- ```
grep i\.,
grep i\\.,
grep "i\."
```
- Gwiazdka \* po symbolu (nawiasie, pojedynczym znaku) nazywana jest domknięciem Kleene'a i oznacza zero lub więcej wystąpień poprzedzającego wyrażenia.
- Znak zapytania ? -zero lub jedno wystąpienie, plus + co najmniej jedno wystąpienie.

## Przykłady

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuer non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.

- `egrep "( ni..)+"`

## Przykłady

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices ac massa. Quisque non makopytko vel orci efficitur posuer non vitae erat. Vivamus eleifend lacus eget pulvinar sollicitudin. Sed ac neque ut tortor gravida mollis vitae at eros. Proin dictum placerat nibh, a fringilla ipsum tincidunt auctor. Vivamus non vulputate quam.

- `egrep "( ni..)+"`
- Vivamus mi nibh, malesuada non dapibus et, tristique at dui. Sed nisl nisl, fringilla ac tristique nec, ultrices at eros. Proin dictum placerat nibh, a fringilla ipsum



## Wyrażenia regularne

- [znaki] - alternatywa znaków.  $[abc]$  oznacza a, b lub c, jak i  $[a-c]$ . A także  $(a | b | c)$ , bo | to prawdziwa alternatywa.  $[\sim a-c]$  wszystkie znaki oprócz tych z zestawu.

## Wyrażenia regularne

- [znaki] - alternatywa znaków.  $[abc]$  oznacza a, b lub c, jak i  $[a-c]$ . A także  $(a | b | c)$ , bo | to prawdziwa alternatywa.  $[\sim a-c]$  wszystkie znaki oprócz tych z zestawu.
- ( i ) grupuje symbole do ich późniejszego wykorzystania.

## Wyrażenia regularne

- [znaki] - alternatywa znaków.  $[abc]$  oznacza a, b lub c, jak i  $[a-c]$ . A także  $(a | b | c)$ , bo | to prawdziwa alternatywa.  $[\hat{a-c}]$  wszystkie znaki oprócz tych z zestawu.
- ( i ) grupuje symbole do ich późniejszego wykorzystania.
- Daszek  $\hat{\phantom{a}}$  oznacza początek wiersza, dolar  $\$$  oznacza koniec wiersza.

In logic-based approaches to multi-agent systems (MAS) there is a long tradition concerned with the development and use of formalisms aimed at expressing the strategic abilities of agents in a system. This interest goes back to earlier work in philosophical logic where formalisms such as STIT have been developed to analyse what agents can bring about in a multi-agent system~\cite{Belnap90}. Against this context the framework of Alternating-Time Temporal Logic (ATL) was developed in the the late 90s in theoretical computer science to reason about paths representing particular outcomes in games~\cite{Alur+02a}.

## grupowanie

```
egrep "[^a-zA-Z]([a-zA-Z]+)[^a-zA-Z]*\1[^a-zA-Z]"
```

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

sed s/nie/tak/



## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

```
sed -r s/\([a-zA-Z]\+\)\1/\1\1\1\1/g
```

## O modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

```
sed -r s/\([a-zA-Z]\+\)\1/\1\1\1\1/g
```

Bardziej zaawansowane rzeczy (zwłaszcza filtrowanie) -> awk (i jego odmiany: gawk, tawk, nawk, mawk, jawk, ...).

# Pisanie wyrażeń regularnych

- liczba naturalna? całkowita?

## Pisanie wyrażeń regularnych

- liczba naturalna? całkowita?
- wzmianka w tekście tweeta?

## Pisanie wyrażeń regularnych

- liczba naturalna? całkowita?
- wzmianka w tekście tweeta?
- e-mail? RFC 5322



## Pisanie wyrażeń regularnych

- liczba naturalna? całkowita?
- wzmianka w tekście tweeta?
- e-mail? RFC 5322

## Pisanie wyrażeń regularnych

- liczba naturalna? całkowita?
- wzmianka w tekście tweeta?
- e-mail? RFC 5322

`local-part@domain`

uppercase and lowercase Latin letters A to Z and a to z

digits 0 to 9

printable characters `!#$%&'*+,-/=?^_`{|}~`

dot `.`, provided that it is not the first or last character

and provided also that it does not appear consecutively

(e.g., `John..Doe@example.com` is not allowed). [5]