

Osnove korištenja operacijskog sustava Linux

05. Zamjenski znakovi i regularni izrazi

Leonard Volarić Horvat
Sebastian Dovičin
Nositelj: dr. sc. Stjepan Groš

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

04.12.2017

Sadržaj

1 Regularni izrazi

2 sed

Regularni izrazi

► Primjer

Izlistati sve datoteke u `/usr/bin` direktoriju koje započinju s `ab`, `bi` ili `ci`

- Vrlo teško sa zamjenskim znakovima

Regularni izrazi

- Korištenje znakova i operatora te pravila regularnih izraza za obradu teksta, pretragu, leksičku analizu, ...
- Moćno i kompleksno proširenje zamjenskih znakova

Regularni izrazi

Sintaksa

Definiranje traženog niza

`x` Znak `x`

`^` Početak reda

`$` Kraj reda

`.` Bilo koji znak

`[xy]` Bilo koji znak u setu

`[^xy]` Invertirani set - Bilo koji znak koji nije u setu

`\[` Znak `[` se interpretira kao traženi simbol,
a ne kao dio sintakse regularnog izraza

`|` ILL operator

Regularni izrazi

Sintaksa

Modifikator broja pojavljivanja niza

x^* Izraz x se pojavljuje bilo koji broj puta ili nijednom

x^+ Izraz x se pojavljuje najmanje jednom

$x^?$ Izraz x se pojavljuje točno jednom ili nijednom

$x\{n\}$ Izraz x se pojavljuje točno n puta

$x\{n,m\}$ Izraz x se pojavljuje $n \leq i \leq m$ puta

Regularni izrazi

► Primjer 1

Regularan izraz koji će zamjenjivati niz znakova `foo` koji se nalazi na početku nekog retka datoteke

```
^foo
```

Regularan izraz koji će zamjenjivati niz znakova `sigh` koji se nalazi na kraju nekog retka datoteke

```
sigh$
```

Regularni izrazi

► Primjer 2

Linija koja ne završava sa slovom A

```
.*[^aA]$
```

Cijeli broj

```
[0-9]+
```

Varijabla u C-u

```
[a-zA-Z]([a-zA-Z0-9])*
```

► Primjer 3

Linija koja završava s tekstovima conf ili log

```
.*[conf|log]$
```

Linija u kojoj postoji troznamenkasti ili četveroznamenkasti broj

```
.*[0-9]{3,4}.*
```

Regularni izrazi

- ▶ Prethodnim tablicama opisan je **prošireni regex**
- ▶ **Osnovni regex** ne sadrži modifikatore broja pojavljivanja niza niti operator `||`
- ▶ Regularne izraze prihvaća mnoštvo programa
- ▶ Mi ćemo se pozabaviti naredbom `grep`
 - `grep` Koristi osnovni regex
 - `egrep` Koristi prošireni regex

Regularni izrazi

- ▶ Pretpostavite kako je niz znakova s desne strane upisan u datoteku regular
- ▶ Odredite koji izlaz će dati sljedeći niz naredbi

```
egrep a.e regular  
egrep a.+e regular  
egrep ab*e regular  
egrep "(ab|cd)e" regular  
egrep "(a|c).+e$" regular
```

```
abe  
abbe  
cde  
45a678  
ae  
cababb  
12345
```

Regularni izrazi

Sintaksa

Klase znakova

- ▶ `[[:alnum:]]` znakovi abecede ili brojevi. A-Za-z0-9
- ▶ `[[:alpha:]]` znakovi abecede. A-Za-z
- ▶ `[[:blank:]]` praznina ili tabularni znak
- ▶ `[[:cntrl:]]` kontrolni znakovi
- ▶ `[[:digit:]]` brojevi. 0-9
- ▶ `[[:lower:]]` mala slova abecede. a-z
- ▶ `[[:upper:]]` velika slova abecede. A-Z
- ▶ `[[:space:]]` praznina
- ▶ `[[:print:]]` znakovi koji mogu biti ispisani. Podrazumijeva znakove u ASCII rasponu 32 – 126 uključujući prazan znak.

Regularni izrazi

► Primjer 4

Izlistajte linije iz datoteke `file.txt` koje sadrže samo brojeve

- Prvi način
\$ `grep [0-9]+ file.txt`
- Drugi način
\$ `grep [[:digit:]]+ file.txt`

Regularni izrazi

Sintaksa

Označeni podizraz

- ▶ Omogućuje korištenje nađenog izraza u daljnjem radu neke naredbe
- ▶ Izraz se označava zatvaranjem kriterija u () zagrade
- ▶ Nađeni izraz se koristi pozivanjem \1 u naredbi, gdje je 1 redni broj označenog podizraza

sed

- ▶ **Stream EDitor**
- ▶ Služi za filtriranje i transformaciju teksta
- ▶ Naredba vrlo širokih mogućnosti
- ▶ Suštinski jednostavan, ali dokumentacija nije baš *user-friendly*
- ▶ Nekoliko naredbi, ali daleko najčešća **s** - *substitute*
 - **Primjer 5**
Naredba sed za filtriranje i obradu teksta
 - `$ sed -r "s/windows/linux/" file.txt`

sed - primjeri

- ▶ Nekoliko primjera:

- ▶ **Primjer 6**

- `$ sed -r "s/windows/linux/" file.txt`

- ▶ mijenja **samo prvo javljanje** riječi "windows" u "linux" u jednom retku
- ▶ navodnici nisu neophodni, ali su poželjni i dobra praksa
- ▶ prvi znak iza slova "s" je delimiter - može biti bilo što, ali najčešće /

► Nadalje:

► **Primjer 7**

- `$ sed -r "s/^Tekst ([[digit:]])\. reda$/Promijenili smo \1. red/" file.txt`

► korištenjem `\1 (\2, ...)` može se sačuvati i ponovno iskoristiti određeni dio reda koji odgovara patternu

► **Primjer 8 - zamjena redoslijeda dvije riječi**

- `$ sed -r 's/([a-z]+) ([a-z]+)/\2 \1/' file.txt`

► ako se nakon zadnjeg delimitera doda "g", on će zamijeniti sva pojavljivanja te riječi u istom retku

Literatura

`man 7 regex`

`man grep`

<http://www.grymoire.com/Unix/Sed.html>

<http://www.regexr.com>

<http://regexone.com>

<http://regex.learncodethehardway.org/book/>

<http://www.rexegg.com>

<http://www.regular-expressions.info/conditional.html>

http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_04.html