

Sed

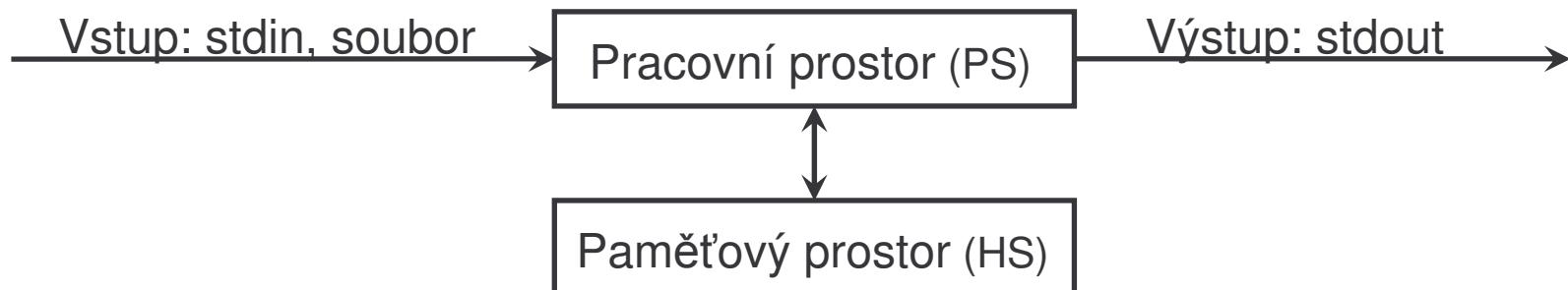
O co jde
Regulární výrazy
Příkazy
Adresování
Příklady

O co jde

- Stream editor
- textový editor vhodný pro dávkové zpracování textu
- je to filtr
- vstup: soubor `sed [-n] [-e script] [-f scriptFile] [soubor]`
`stdin cat soubor | sed [-n] [-e script] [-f scriptFile]`
- kde:
 - n potlačení kopírování vstupu na výstup
 - e sedovský příkaz zadaný z řádky
 - f načte soubor se sedevskými příkazy
- výstup: vždy na stdout => **vstupní soubor se nemění**, pozor na přesměrování!
- 2 režimy práce:
 - standardně se vše kopíruje ze vstupu na výstup
 - n nic se nekopíruje
- pro práci nutná znalost regulárních výrazů

O co jde (pracovní cyklus)

- 1) načtena řádka ze zdrojového souboru a interně očíslována
- 2) řádka vložena do pracovního prostoru
- 3) na obsah pracovního prostoru aplikován sedscript (posloupnost sedovských příkazů)
- 4) obsah pracovního prostoru poslán na standardní výstup



Regulární výrazy

- vzor porovnávaný s daným textem
- regulární výraz používá speciální znaky v jiném významu než shell
([,],., *, \$)

Jednoznačkové

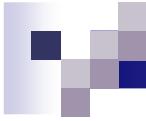
Konstrukce	se porovná
znak	sám se sebou
\spec_znak	se speciálním znakem (., *, [,], \) jako takovým.
.	s jedním libovolným znakem.
[znaky]	s libovolným jedním znakem z uvedených
[^znaky]	s libovolným jedním znakem různým od uvedených
-	uvedené mezi znaky znamená interval

Regulární výrazy (Víceznakové)

Konstrukce	se porovná
$<1znRE>^*$	s libovolným (i nulovým) opakováním předchozího jednoznakového RE. Je-li více možností, vezme se první porovnání zleva co nejdelší.
$<1znRE>\{m\}$	s právě m opakováními jednoznakového RE (m je číslo 1 až 255)
$<1znRE>\{m,\}$	s nejméně m opakováními jednoznakového RE
$<1znRE>\{m,n\}$	s m až n opakováními jednoznakového RE
$\langle\langle RE \rangle\rangle$	s odpovídajícím řetězem od začátku slova
$\rangle\langle RE \rangle$	s odpovídajícím řetězem na konci slova
$^<RE>$	odpovídajícím řetězem od začátku řádky
$RE>\$$	odpovídajícím řetězem na konci řádky

Příkazy

- **d** (delete) - vymaže řádku
- **p** (print) - opíše řádku na výstup
- **r soubor** - načte uvedený soubor a pošle ho na výstup
- **s/RE/náhrada/volby** - nahradí text porovnaný s **RE**
v **náhrada** se může vyskytnout:
 - & - text, se kterým se porovnalo **RE**
 - \n** (**n** je číslice) - řetězec se kterým se porovnal **n**-tý výskyt
\(REn\) v **RE**
- **:** návěští návěští
- **b**návěští - nepodmíněný skok na návěští
- **t**návěští - podmíněný skok, pokud poslední náhrada **s** byla úspěšná
- Příkazy pro přesouvání mezi PS a HS: **h,H,g,G,x**
- **#** komentář (**#n** na první řádce to samé jako **-n**)
- spousta dalších



Adresování

- 1) bez adresy příkaz proveden pro každou řádku vstupu
sed -n -e p t.txt
 - 2) 1 adresa proveden pouze pro tuto řádku
sed -e /aa/d t.txt
 - 3) Rozsah adres reaguje na řádky od – do (včetně)
sed -e /aa/world/d t.txt

Adresou může být:

- číslo řádku
 - regulární výraz (RE)
 - \$ (poslední řádka souboru)

Příklady

```
$ cat t.txt
```

```
hello world!!!
```

```
how are you?
```

```
aa bbbb
```

```
abc
```

```
aaaa
```

```
aa bbb dd
```

```
qwerty
```

```
$ sed -e /a/d t.txt
```

```
hello world!!!
```

```
qwerty
```

```
$ sed -n -e /a/p t.txt
```

```
how are you?
```

```
aa bbbb
```

```
abc
```

```
aaaa
```

```
aa bbb dd
```

```
$ cat s.sed
```

```
#n
```

```
# sed script
```

```
s/[ab]$/end/p
```

```
$ sed -f s.sed t.txt
```

```
aa bbbend
```

```
aaaend
```



Další informace

- unix.felk.cvut.cz/unix/download/predn_05.pdf
- www.root.cz/clanky/neco-malo-o-sedu/
- www.root.cz/clanky/neco-malo-o-sedu-priklady-pouziti/