

Základy práce s nástroji grep a sed

Tomáš Kühr



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

- filtruje řádky vstupu odpovídající zadanému vzoru
- ve vzoru mohou být využity konstrukty rozšířených regulárních výrazů
- vytvořil ho Ken Thompson v roce 1974

Nejdůležitější přepínače

- zadaný vzor odpovídá celému slovu (přepínač -w)
- rekurzivní prohledávání adresářů (přepínač -r)
- výpis doplněn o čísla řádků (přepínač -n)
- výpis počtu řádků odpovídajících vzoru (přepínač -c)
- ignorování velikosti písmen (přepínač -i)
- použití zjednodušeného rozšířených možností „regulárních výrazů“ (přepínač -E)

- 1 libovolný znak (znak `.` ve vzoru)
- nepovinný výskyt znaku (`\?` za daným znakem)
- libovolný (i nulový) počet opakování znaku (`*` za daným znakem)
- nenulový počet opakování znaku (`\+` za daným znakem)
- přesně daný počet opakování (`\{počet\}` za daným znakem)
- počet opakování v daném rozsahu (`\{od, do\}` za daným znakem)
- alespoň daný počet opakování (`\{od, \}` za daným znakem)
- libovolný znak z množiny (`[výčet]` nebo `[od-do]`)
- libovolný znak mimo množinu (`[^výčet]` nebo `[^od-do]`)
- lze použít i mnoho předdefinovaných skupin znaků (např. `[:alpha:]`)
- definování alternativ pomocí logického spojky „nebo“ (znaky `|`)
- možnost pracovat se začátky řádků (znak `^`)
- možnost pracovat s konci řádků (znak `$`)
- seskupování znaků pro potřeby výše zmíněných operací pomocí `\(` a `\)`

Příklady použití

- Výběr řádků obsahujících řetězec
`grep kuhrtoma /etc/passwd`
`ls -l ~/. | grep "Feb 22"`
- Hledání řádků obsahujících vzor jako celé slovo
`ps aux | grep -w "ps"`
- Vypsání pouze počtu řádku
`ps aux | grep -cw "ps"`
- Hledání řádků se vzorem na začátku
`ls -l ~/zalozy/. | grep "^total"`
- Hledání řádků končících číslicí 0 až 4
`grep "[0-4]$" books.txt`
- Hledání řádků končících 2 nebo 3 číslicemi
`grep -E "[[:digit:]]{2,3} $" books.txt`
- Hledání řádků obsahujících „process“ nebo „file“
`grep -E 'Martin|Tolkien' books.txt`

- vytvořen 1974 v Bellových laboratořích
- autor Lee E. McMahon
- sloužící k aplikaci nejrůznějších transformací na sekvenci (proud) textových dat
 - odstranění řádku
 - záměna řádku
 - vložení řádku
 - vyhledání a nahrazení vzoru
 - prohození řetězců odpovídajících vzorům
 - záměny znaků
- mnoho možností, jak specifikovat řádky, na kterých se mají transformace provést
- vstupním proudem je obvykle soubor nebo výstup jiného programu
- výstup může být uložen do souboru nebo použit jako vstup jiného programu

1 Read

- načte se jeden řádek vstupního proudu (souboru)
- existují ale i možnosti, jak pracovat s více řádky najednou

2 Execute

- zjistí se, zda se má daný příkaz (transformace) na načtený řádek použít
 - podle čísla řádku
 - podle nalezení/nenalezení vzoru na řádku
 - některé příkazy se provádějí vždy
- pokud má být řádek transformován, provede se změna v bufferu
- zpracovávaných příkazů může být větší množství, postupně se provádějí všechny

3 Display

- upravený řádek se pošle do výstupního proudu

Příkazy zadané přímo do promptu

- pokud je příkazů malé množství a jsou spíše jednodušší
- pokud si nechceme příkazy uložit pro pozdější použití
- příklad: `sed -e '1d' -e '2d' -e '5d' books.txt`

Příkazy zadané v souboru

- pokud je příkazů více nebo jsou složitější
- pokud chceme stejné příkazy používat někdy v budoucnu
- příklad: `sed -f commands.txt books.txt`
- pokud příkazy v souboru doplníme prvním řádkem: `#!/bin/sed -f`
- a nastavíme právo spouštět daný soubor
- lze použít také `./script.txt books.txt`

Nejobvyklejší přepínače

- předání více příkazů z promptu (přepínač `-e`)
- zpracování příkazů ze souboru (přepínač `-f`)
- pouze výpis řádků, u kterých je uveden příkaz pro tisk (přepínač `-n`)

Číslem řádku

- uvedením čísla řádku před příkazem
- příklad: `sed -n '3p' books.txt`

Rozsahem čísel řádků

- uvedením čísla prvního a posledního řádku oddělených čárkou
- příklad: `sed -n '2,5 p' books.txt`
- poslední řádek je možné vyjádřit pomocí `$`
- příklad: `sed -n '3,$ p' books.txt`

Počtem řádků a číslem prvního

- číslo posledního řádku výčtu je možné vyjádřit i relativně k prvnímu (pomocí `+`)
- příklad: `sed -n '2,+3 p' books.txt`

Každý x-tý řádek

- lze zpracovávat také každý x-tý řádek počínaje zadaným řádkem
- příklad: `sed -n '2~2 p' books.txt`

Řádky s výskytem vzoru

- vzorem může být textový řetězec, ale i regulární výraz (viz dále)
- vzor zapisujeme mezi lomítka
- příklad: `sed -n '/Paulo/ p' books.txt`

Řádky od jednoho vzoru po druhý vzor

- zapisujeme podobně jako při rozsahu čísel řádků
- příklad: `sed -n '/Alchemist/,/Pilgrimage/ p' books.txt`
- vyjádření začátku/konce úseku vzorem lze kombinovat s vyjádřením konce/začátku číslem řádku
- příklady:
`sed -n '/Alchemist/,5 p' books.txt`
`sed -n '3,/Pilgrimage/ p' books.txt`
`sed -n '/Alchemist/,$ p' books.txt`
`sed -n '/Alchemist/,+2 p' books.txt`

Tisk řádku

- pomocí příkazu `p`
- pokud nebyl `sed` spuštěn s přepínačem `-n`, duplikace
- příklad: `sed -e 'p' -e 'p' books.txt`

Smazání řádku

- pomocí příkazu `d`
- příklad: `sed '2,4 d' books.txt`

Ukončení programu

- pomocí příkazu `q`
- pokud chceme provádět úpravy pouze do určitého místa textu
- příklad: `sed '/The Alchemist/ q' books.txt`

Výměny znaků

- pomocí příkazu `y`
- za příkazem se uvedou vzory a obrazy znaků oddělené lomítky
- příklad: `sed '3,5 y/aeiouy/AEIOUY/' books.txt`

Vložení řádku za daný řádek

- pomocí příkazu a
- příklad: `sed '$ a 7) Adultry, Paulo Coelho, 234' books.txt`

Vložení řádku před daný řádek

- pomocí příkazu i
- příklad: `sed '/The/ i 7) Adultry, Paulo Coelho, 234' books.txt`

Nahrazení řádku jiným

- pomocí příkazu c
- příklad: `sed '3 c 3) Adultry, Paulo Coelho, 324' books.txt`

Možnost vkládat i více řádků

- před, za či místo jednoho řádku
- ve vkládaném řetězci použijeme speciální znak `\n`
- pokud naopak ve skriptu dělíme příkaz na více řádků použijeme `\`
- příklad: `sed '$ a 7) Adultry, Paulo Coelho, 234\n8) Eleven Minutes, \n Paulo Coelho, 304' books.txt`

Vložení čísla řádku

- pomocí příkazu =
- číslo řádku se vloží do výstupu před tento řádek (pokud se zobrazuje)
- příklady:

```
sed '/Paulo/ =' books.txt
```

```
sed -n '$ =' books.txt
```

Inverze výběru

- pomocí příkazu !
- příklad:

```
sed '1~3 !d' books.txt
```

Seskupování

- pomocí složených závorek
- umožňuje snadno definovat složitější výběry
- příklad:

```
sed -n '1~3 {/The/ p}' books.txt
```

- 1 libovolný znak (znak `.` ve vzoru)
- nepovinný výskyt znaku (`\?` za daným znakem)
- libovolný (i nulový) počet opakování znaku (`*` za daným znakem)
- nenulový počet opakování znaku (`\+` za daným znakem)
- přesně daný počet opakování (`\{počet\}` za daným znakem)
- počet opakování v daném rozsahu (`\{od, do\}` za daným znakem)
- alespoň daný počet opakování (`\{od, \}` za daným znakem)
- libovolný znak z množiny (`[výčet]` nebo `[od-do]`)
- libovolný znak mimo množinu (`[^výčet]` nebo `[^od-do]`)
- lze použít i mnoho předdefinovaných skupin znaků (např. `[:alpha:]`)
- definování alternativ pomocí logického spojky „nebo“ (znaky `|`)
- možnost pracovat se začátky řádků (znak `^`)
- možnost pracovat s konci řádků (znak `$`)
- seskupování znaků pro potřeby výše zmíněných operací pomocí `\(` a `\)`

- `sed -n '/^ The/ p' books.txt`
- `sed -n '/^ .) The/ p' books.txt`
- `sed -n '/[468]$/ p' books.txt`
- `sed -n '/\(Tolkien\|Martin\) ,/ p' books.txt`
- `sed -n '/[[[:digit:]]\{4\}/ p' books.txt`
- `sed -nr '/[[[:digit:]]{4}/ p' books.txt`
- `sed -n '/100\?/ p' numbers.txt`
- `sed -nr '/100?$/ p' numbers.txt`
- `sed -nr '/10{5}$/ p' numbers.txt`
- `sed -nr '/10{5,7}$/ p' numbers.txt`
- `sed -nr '/10{5,}$ / p' numbers.txt`
- `sed -nr '/1(00)*$/ p' numbers.txt`

- 1 Pomocí nástroje grep zobrazte pouze ty řádky vstupního souboru, které obsahují jednoduché matematické formule – tj. vždy jeden operátor = , libovolné základní matematické operace (+, -, * /) a proměnné (znaky a až z). (1 bod)
- 2 Vytvořte sed skript, který v textovém souboru najde všechny řádky mezi značkami BEGIN a END, které obsahují nějaké desetinné číslo. Můžete předpokládat, že BEGIN a END jsou na samostatných řádcích. (2 body)