

F4270 UNIX, počítačové sítě (jaro 2016), Cvičení 6

Příklady: sed, ed, regulární výrazy. Prostudujte si man stránky k těmto příkazům.

- A1** Ze souboru calories.csv vyberte řádky, jejichž množství je měřeno jednotkou „Piece“ nebo „Cake“.
- ```
[guy@nix ~]$ grep -e '^[^;]*;[[:digit:]]* Piece;' \
-e '^[^;]*;[[:digit:]]* Cake;' calories.csv
[guy@nix ~]$ grep -E '^[^;]*;[[:digit:]]* (Piece|Cake);' calories.csv
```
- A2** Z výpisu ls -l vyberte řádky, u nichž má vlastník nastavena stejná práva jako skupina i jako ostatní. (Případně si odpovídající soubor například v /tmp vytvořte).
- ```
[guy@nix ~]$ ls -l | grep '^\(...)\|1\|1'
```
- A3** Ze souboru kodyzemi_cz.csv vyberte řádky, v nichž dvouznakový kód je shodný s prvními dvěma znaky trojznakového kódu země.
- ```
[guy@nix ~]$ grep '^[[:digit:]]*;\(..\);\1;' kodyzemi_cz.csv
```
- A4** Ze souboru kodyzemi\_cz.csv vyberte naopak řádky, v nichž dvouznakový kód není shodný s prvními dvěma znaky trojznakového kódu země.
- ```
[guy@nix ~]$ grep -v '^[[:digit:]]*;\(..\);\1;' kodyzemi_cz.csv
```
- A5** Pomocí sedu vyberte z výpisu ls -l jen sloupec s velikostí souboru.
- ```
[guy@nix ~]$ ls -l | sed '1d; s/^\([^]* *\)\{5\} .*/\1/'
```
- Řešení v edu:
- ```
#!/bin/sh  
# Všimněte si, že řetězec KONEC je uvozen zpětným lomítkem, to  
# proto, aby shell neprováděl expanzi v textu, který za ním píšeme.  
ls -l > /tmp/lsl.$$  
ed /tmp/lsl.$$<<\KONEC  
2,$g/^/s/^\([^ ]* *\)\{5\} .*/\1/  
%p  
Q  
KONEC  
rm /tmp/lsl.$$
```
- A6** Pomocí sedu přetvořte řádky /etc/passwd do podoby uid (login).
- ```
[guy@nix ~]$ sed 's/^\([^:]*\):[~:]*:\([^:]*\).*/\2 (\1)/' /etc/passwd
```
- Řešení v edu:
- ```
#!/bin/sh  
# Pokud chceme na konci výsledek uložit do souboru, pak tam bude w a q.  
cp /etc/passwd /tmp/passwd.$$  
ed /tmp/passwd.$$<<\KONEC  
g/^/s/^\([^:]*\):[~:]*:\([^:]*\).*/\2 (\1)/  
%p  
Q  
KONEC  
rm /tmp/passwd.$$
```
- A7** Z výstupu nl /etc/passwd vypíšte jen sudé řádky.
- ```
[guy@nix ~]$ nl /etc/passwd | sed -n 'n; p'
```
- Řešení v edu:
- ```
#!/bin/sh  
# Příkaz g provádí následující příkazy na všechny řádky.  
# V rámci něj:  
# 1) +1p vypíše následující řádku a  
# 2) .s/^// odznačí (následující) řádku tak, že ji g vynechá.  
# poslední řádku vynecháváme (předpokládáme, že soubor obsahuje aspoň  
# dvě řádky), protože u ní by +1 lezlo ven ze souboru.
```

```
#
# Kdybychom použili s///, tak místo prázdného regexpu by g doplnilo
# svůj regexp a došlo by k vymazání části sudých řádek. Pokud se neukládají
# změny, tak to nevadí.
```

```
nl /etc/passwd > /tmp/passwd.$$
ed /tmp/passwd<<\KONEC
1,$-1g/~/+1p\
.s/~/
Q
KONEC
rm /tmp/passwd.$$
```

A8 Na začátky lichých řádek souboru `/etc/passwd` vložte znak „l“, na začátky sudých řádek vložte znak „s“.

```
#!/usr/bin/sed -f
# Toto je skript pro sed, použití pomocí sed -f. Volání: sed -f <skript> /etc/passwd
s/~/l/
n
s/~/s/
```

Řešení v edu:

```
#!/bin/sh
# První příkaz přidá znak „l\ na začátek každé řádky.
# Druhý příkaz změní znak „s\ na začátku každé sudé řádky.
# Poté je vypsán výsledek a ed ukončen bez uložení změn.
ed /etc/passwd <<\KONEC
g/~/s/~/l/
1,$-1g/~/+1s/~/s/
%p
Q
KONEC
```

A9 Na začátky desáté až dvacáté řádky souboru `/etc/passwd` přidejte znak #.

```
[guy@nix ~]$ sed '10,20s/~/#/' /etc/passwd
```

Řešení v edu:

```
#!/bin/sh
# Změna se neukládá (do /etc/passwd by to nebyl dobrý nápad) a
# výsledek se jenom vypíše na standardní výstup.
ed /etc/passwd <<\KONEC
10,20s/~/#/
%p
Q
KONEC
```

A10 Před první řádku vstupního souboru vložte řádku s `#!/bin/sh`.

```
#!/usr/bin/sed -f
# Toto je skript pro sed, použití pomocí sed -f
1i\
#!/bin/sh
```

Řešení v edu:

```
ed vstupnisoubor <<\KONEC
1i
#!/bin/sh
.
w
q
KONEC
```

- A11** Napište skript pro sed, který odstraní ze shelového skriptu komentáře, přičemž
- Pokud jde o první řádku souboru a pokud tato řádka má formát `^#!.*`, pak tento komentář neodstraňujte.
 - Řádky, kde před `#` není nic než mezery, odstraňte úplně.
 - Pokud se před `#` vyskytují nějaké jiné znaky, odstraňte z řádky vše od prvního výskytu znaku `#` (včetně) do konce řádky.

```
#!/usr/bin/sed -f
# Přeskočení první řádky, pokud jde o komentář typu #!
1{
/^#!/n
}
# Smazání řádek, kde před # nic není
/^[[:space:]]*#/d
# Odstranění komentáře z ostatních řádek (ignoruje fakt, že # se může
# vyskytovat i v rámci řetězce v uvozovkách a pod., kdybychom toto
# chtěli kontrolovat, bylo by to výrazně komplikovanější).
s/#.*$//
```

Řešení v edu:

```
#!/bin/sh
# Nejprve vyřídíme vše kromě prvního řádku.
# - První příkaz smaže řádky, které před # nic nemají.
# - Druhý příkaz odstraní z ostatních řádků vše od # dál.
# Poté vyřešíme první řádek.
# - Nejprve vymažeme 1. řádek, pokud obsahuje komentář, který před # nic
# nemá a který napak za # má nějaký znak, který není !.
# - Poté smažeme 1. řádek, pokud obsahuje jen #, za nímž ani před nímž
# nic není. Pokud předchozí příkaz proběhl úspěšně, dostane se sem 2.
# řádek, ale z něj jsme již komentáře odstranili, a tak se nám to
# nemůže splést a další příkazy se pro něj nevykonaly. Z tohoto
# důvodu jsme také nejprve řešili řádky od 2. do posledního.
# - Nakonec z prvního řádku odstraníme vše od prvního # do konce, pokud
# nemá #! na začátku.
# Všude je nutné používat g nebo v, protože jinak by s při neúspěšné
# substituci hlásilo chybu.
ed vstupnisoubor.sh <<\KONEC
2,$g/^[[:space:]]*#/d
2,$g/#/s/#.*//
1g/^[[:space:]]*#[^!]/d
1g/^[[:space:]]*#$/d
1v/^#!/s/#.*$//
w
q
KONEC
```

- A12** Napište skript pro sed, který otočí pořadí řádek na vstupu.

```
[guy@nix ~]$ nl /etc/passwd | sed -n 'G; $p; h'
```

Řešení v edu:

```
#!/bin/sh
# Zpětné lomítko je za příkazem i nutné proto, že jde o příkaz v rámci
# příkazu g, jinak by tam nebylo potřeba.
ed /etc/passwd <<\KONEC
g/^/m0
%p
Q
KONEC
```

- A13** Mezi každé dvě řádky souboru `/etc/passwd` vložte prázdnou řádku.

```
sed '2,$i\  
' /etc/passwd
```

Nebo:

```
sed '$b; a\  
' /etc/passwd
```

Řešení v edu:

```
ed /etc/passwd <<\KONEC  
2,$g/^/i\  
\  
.  
%p  
Q  
KONEC
```

- A14** Napište skript pro `sed`, který na každé řádce nahradí jednořádkové komentáře typu `/* ... */` komentářem typu `//...`, uvažte, že na řádce může být těch jednořádkových komentářů víc a že kolem komentářů je i obyčejný text. Z jednoho řádku tak vznikne víc řádek, protože za každým komentářem `//` musíte odřádkovat.

```
#!/usr/bin/sed -f  
# Toto je skript pro sed, použití pomocí sed -f  
: zacatek  
s%\(.*\)/\*\(.*\)\*/\(.*)%\1//\2\  
\3%  
t zacatek
```