A7B38UOS Úvod do operačních systémů

6. Cvičení

Příkazy sed a awk

sed

sed [přepínače] 'příkaz' [soubory]

sed [přepínače] -f skript [soubory]

Stream editor - edituje neinteraktivně jeden nebo více souborů.

Jednotlivé řádky jsou čteny ze standardního vstupu nebo ze souboru, provádí nad nimi jednotlivé příkazy a vypíší se na standardní výstup.

-n potlačí implicitní kopírování vstupu na výstup
(to co se má vytisknou musí být explicitně určeno příkazem p)
-f skript (soubor skript musí obsahovat seznam příkazů)
-r pracuje s rozšířenými regulárními výrazy

Formát příkazu: [adresa1 [,adresa2]] příkaz [parametry]

Příkazy: p(rint) vypíše řádku na výstup
d(elete) zruší řádku
s/RE1/RE2/volby nahradí text, který odpovídá vzoru RE1, řetězcem RE2
q(uit) ukončí běh

sed – adresy, print/delete I

Výpis prvních 10 řádků (head)

cat -n /etc/passwd | sed '11,$d'

cat -n /etc/passwd | sed -n '1,10p'

Výpis posledních 10 řádků (tail)

sed -n $((`wc -l </etc/passwd`-10)),\$p /etc/passwd

Výpis řádků obsahujících RE (grep)

sed -n /root/p /etc/group

Výpis řádků neobsahujících RE (grep -v)

sed /root/d /etc/group

sed – adresy, print/delete II

Výpis řádků od „case“ do „esac“

sed -n '/^case/,/^esac/p' /etc/init.d/cups

Výpis začátku souboru do prvního prázdného řádku

sed '/^$/,$d' /etc/init.d/cups

sed -n '1,/^$/p' /etc/init.d/cups

sed '/^$/q' /etc/init.d/cups

sed – náhrady v textu I

Náhrada prvního znaku

```
ls -l /etc \
    | sed 's/^-/file:/' \
    | sed 's/^d/ dir:/' \
    | sed 's/^l/link:/' \
    | sed 's/^p/pipe:/'

ls -l /etc | sed 's/^-/file: /;s/^d/ dir: /;
s/^l/link: /;s/^p/pipe: /'
```

sed – náhrady v textu II

Náhrada prvního slova man na řádku za !!man!!

man man | sed 's/\<man\>/!!man!!/'

Náhrada druhého slova man na řádku za !!man!!

man man | sed 's/\<man\>/!!man!!/2'

Náhrada všech slov man na řádku za !!man!!

man man | sed 's/\<man\>/!!man!!/g'

Výpis pouze těch řádků, kde došlo k náhradě man za !!man!!

man man | sed -n 's/\<man\>/!!man!!/gp'

... kde došlo k náhradě bez ohledu na velikost písmen

man man | sed -n 's/\<[Mm][Aa][Nn]\>/!!man!!/gp'

sed – náhrady textu +

„Obalení“ čísel do <number: >

man man | sed 's/\([0-9][0-9]*\)/<number: \1>/g'

Změna formátu výpisu linků (pouze linků)

ls -l /etc | sed -n \

    's/^l.* \([^]*\) -> \([^]*\)/Link: \1 (\2)/p'

awk a nawk I

awk [přepínače] [program] [prom=hod...] [soubory]

Programovatelný filtr vytvořený autory: Aho, Weinberger, Kernighan.

Původní verze se jmenovala awk, rozšířená verze pak nawk (na některých systémech lze rozšířenou verzi najít pod původním názvem awk).

Jednotlivé řádky jsou čteny ze standardního vstupu nebo ze souboru, pokud odpovídají zadanému vzoru, provede se nad nimi příslušný program.

Pokud není vzor specifikován, provede se program nad každou řádkou.

Na řádku se pohlíží jako na posloupnost položek $1, $2,...,$NF ($0 = celá řádka).

Implicitním oddělovačem položek je mezera/tabelátor (lze změnit přepínačem –F nebo proměnnou FS).

Struktura příkazu programu:
[vzor] [{ akce }]

awk - vzory

Typy vzorů:

Vzor	Kdy se provede akce
BEGIN	před zpracováním první řádky ze vstupu
END	po zpracováním poslední řádky ze vstupu
výraz	pro řádky vyhovující danému výrazu
Začátek, konec	od první řádky splňující výraz začátek až do první řádky splňující výraz konec

Typy výrazů:

regulární výraz (ve formátu pro egrep)
logický výraz (0 nebo prázdný řetězec = false, jinak true)

awk – logické výrazy, proměnné

Logické výrazy

- tvořeny pomocí operátorů jako v jazyce C
- relační operátory: >, >=, <, <=, ==, !=, ~, !~ (řetězec odpovídá/neodpovídá danému vzoru)
- matematické operátory: +, -, *, /, %,^, ++, --
- logické operátory: &&, ||, !

Proměnné

- použití je podobné jako v jazyce C
- deklarují se použitím

Některé předdefinované proměnné

- $n hodnota n-té položka z aktuálního řádku ($0 = celá řádka)
- NF počet položek v aktuálním řádku
- NR pořadová číslo aktuální řádky
- FS vstupní oddělovač položek na řádce
- OFS výstupní oddělovač položek na řádce

awk - příklady

Výpis prvních 10 řádků (head)

```
cat -n /etc/passwd | awk 'NR<=10'
cat -n /etc/passwd | awk 'NR==1,NR==10'
```

Výpis posledních 10 řádků (tail)

```
awk NR\>=$((`wc -l </etc/passwd`-10)) /etc/passwd
```

Výpis řádků obsahujících RE (grep)

```
awk /root/ /etc/group
```

Výpis řádků neobsahujících RE (grep -v)

```
awk '!/root/' /etc/group
```

Z výstupu ypcat passwd výběr položek s UID 1000-9999

```
ypcat passwd | awk -F: '$3>=1000 && $3<=9999'
```

awk – podmíněné příkazy, cykly, funkce

Podmíněný příkaz

if (výraz) { příkazy1} [else { příkazy2}]

Cykly

for (i=min; i<=max; i++) { příkazy }

for (j in pole) { příkazy }

while (výraz) { příkazy }

do { příkazy } while (výraz)

break # předčasné ukončení cyklu

continue # předčasné ukončení aktuální iterace cyklu

Předdefinované funkce

printf("řetězec" [,hodnoty])

sin(), sqrt(), log(), exp(),...

length(), match(), split(), substr(), sub(),...

tolower(), toupper(),...

awk - příklady

Výběr a změna pořadí jednotlivých položek /etc/passwd

```
awk 'BEGIN { FS=":"; OFS=":" } { print $3,$1,$5 }' /etc/passwd
```

Nalezení uživatele s nejdelším jménem

```
ypcat passwd | awk -F: '
 {
if (length($5) > max) {
        max=length($5); user=$1; name=$5
}
 }
 END {
print user ":" name " (" max ")"
 }'
```

Výpis uživatelů ve skupině fuse

```
nawk -F: '
 /^fuse/ {
items=split($4,a,",")
for (i=1; i<=items; i++) print a[i]
 }' /etc/group
```

awk - příklady+

Formátovaný výpis uživatelů Korn shellu

```
awk -F: '
 BEGIN {
      print "Uzivatele /bin/sh shellu:"
      for (d=0; d<=60; d++) printf "-"; printf "\n"
 }
 $7 ~ /\/sh/ {
      if (length($1)<=7) tab="\t\t"; else tab="\t"
       print $1 tab $5
       sh++
      }
 END {
      for (d=0; d<=60; d++) printf "-"; printf "\n"
      print "Celkem " sh " z " NR " uzivatelu"
 }' /etc/passwd
```