

Цевки, филтри и регуларни изрази

Цевки

Операторот "|", кој се нарекува цевка (pipe), се користи за поврзувањето на командите. Притоа, излезот од првата команда се користи како влез на втората. Многу честа комбинација е следната:

```
student@os:~$ ls -al | less
```

Имено, ако во именикот има многу датотеки, излезот од `ls` "ќе избега". Командата `less` ќе го преземе тој излез и ќе го прикаже страна по страна.

Со "|" може да се поврзуваат произволен број команди. Таква командна линија се нарекува pipeline. На пример со:

```
student@os:~$ ls -al Lab2 | sort | cat > lista.txt
```

листата на датотеки во именикот `Lab2` ќе се сортира и ќе се пренасочи во датотеката `lista.txt`.

Филтри

Филтрите се команди кои обично вршат трансформации на некоја датотека, на излезот од друга команда или даваат информации за нив. Пример на таква команда е командата `grep`. Таа може да се користи и како самостојна команда или во комбинација со друга. Нејзиниот општ облик е:

`$grep регуларен_израз датотека(и)`

`grep` чита од датотеката(и) или од стандардниот влез и на стандардниот излез ги прикажува редовите кои содржат низа од знаци која одговара на регуларниот израз. Наједноставната форма е кога како регуларен израз се користи низа од знаци. На пример:

```
student@os:~$ grep zdravo poraka
```

ќе ги прикаже сите редови од датотеката `poraka` кои ја содржат низата `zdravo`. Многу почесто `grep` се користи во комбинација со други команди поврзани со цевка. Командата:

```
student@os:~$ cat poraka | grep '^....97'
```

ќе ги прикаже сите редови од `poraka` кои започнуваат со низа од 4 знаци по која следи '97'. Ако наместо `^` се употреби `$` (на крајот) , тогаш ќе се бараат оние редови кои завршуваат на зададената низа од знаци. Со опцијата `-v` се пронаоѓаат сите редови од датотеката кои не содржат одредена низа од знаци. Со:

```
student@os:~$ grep -v zdravo poraka
```

ќе се прикажат сите редови од `poraka` кои не содржат `zdravo`.

Еве еден посложен пример на команда со филтер која многу често се користи:

```
student@os:~$ ls -l | grep '\.txt'| less
```

Резултатот ќе биде листа на сите датотеки во тековниот именик кои завршуваат на наставка `.txt`, излистани страна по страна (со знакот `" \ "` се отстранува специјалното значење на точката, исто важи и за останатите специјални знаци).

Командата `sort` исто така спаѓа во групата на филтрите. Се употребува во вид:

`$sort` опции *датотека* или поврзана со цевка со друга команда.

Некои од опциите на `sort` се:

- f - сортирање по абecedен редослед без разликување на големите и малите букви
- n - сортирање по нумерички вредности
- r - сортирање по опаѓачки редослед
- k x (x е цел број) - сортирање според x-тиот збор во редот

Примери:

```
student@os:~$ ls | sort -f
```

ги сортира датотеките од тековниот именик по абecedен редослед

```
student@os:~$ ls -l | sort -rn -k 5
```

ги сортира датотеките по големина по опаѓачки редослед

```
student@os:~$ who | sort -k 5
```

ги сортира корисниците на системот според нивното време на најавување.

Некогаш сакаме да зададеме низа од повеќе команди одеднаш на командната линија. Тоа може да се реализира со помош на операторот ";" кој се користи на сличен начин како и "|", но не ги поврзува командите меѓу себе.

Регуларни изрази

Регуларните изрази се употребуваат при разни манипулации со текст. Тие претставуваат замена за низи од знаци кои задоволуваат одредени услови.

Еден регуларен израз се состои од оператори кои опишуваат низи од знаци (слично како * (заменува било кои карактери) и ? (заменува еден карактер)). Следните комбинации од оператори можат да се користат за конструкција на регуларни изрази:

. - заменува било кој знак

^ - означува барање од почеток на ред, но некогаш во зависност од контекстот го негира значењето на множество карактери

\$ - означува барање од крај на ред

\c - ако c е специјален знак, се третира како обичен

[c₁–c₂] - било кој знак од интервалот c₁–c₂

[c₁c₂ . . . c_n] - било кој знак од низата c₁c₂...c_n

[^c₁–c₂] - било кој знак кој не припаѓа на интервалот c₁–c₂

[^c₁c₂ . . . c_n] - било кој знак кој не припаѓа на низата c₁c₂...c_n

\< - означува барање од почеток на збор

\> - означува барање од крај на збор

\b - означува барање од почеток или крај на збор

\B - означува барање било кој карактер но не на почеток или крај на збор

За секвенци од ист карактер може и да ги користиме следниве регуларни изрази:

\{n\} - точно n пати повторување на последниот карактер

\{n, \} - n или повеќе пати повторување на последниот карактер

\{n, m\} - минимум n а максимум m пати повторување на последниот карактер

Примери:

a . c - било која низа која почнува на a, завршува на c и има еден знак меѓу нив

[a–zA–Z] - сите афабетски знаци

[0–9] [aA] [^a–z] - било која низа која почнува со цифра, потоа има уште 5 знаци, следи 'a' или 'A', а последниот знак не е мала буква.

sa\{5, 10\}t - било кој збор што почнува со s а завршува со t а измеѓу има помеѓу 5 и 10 карактери 'a'.

\<c.*h\> - било кој збор што почнува на c и завршува на h

Следната команда ги листа сите датотеки кои имаат наставка .cc

```
student@os:~$ ls *.cc
```

додека пак со командата

```
student@os:~$ ls -l [abcdefghijklmnopqrstuvwxyz]* | less
```

ќе се излистаат сите именици и датотеки чии имиња почнуваат на буквите 'a', 'b', 'c', 'x', 'y' или 'z'.

За да ги избришете сите датотеки со име prog и наставка која има еден знак, зададете:

```
student@os:~$ rm prog?
```

додека пак со командата

```
student@os:~$ rm *[^1-9]
```

ќе се избришат сите датотеки чие име не завршува на број

Со командата:

```
student@os:~$ mkdir newdir student@os:~$ cp [a-z]* newdir/
```

ќе ги ископира сите датотеки, чие име започнува со буква во подименикот newdir од тековниот именик.

```
student@os:~$ cp [pP]art[0-9] ../book
```

ќе ги ископира сите датотеки, чие име започнува со p или P, потоа следи низата art и завршува со цифра, во именик book кој се наоѓа едно ниво погоре од она на тековниот именик. Внимавајте именикот book треба да се креира доколку не постои!

```
student@os:~$ mv {ii,[0-9]}*.{txt,doc,jpg} /home/student/lab2
```

ќе се преместат сите датотеки чие име започнува на ii или некоја цифра, па следува низа од карактери и завршува со .txt или .doc или .jpg во именикот student/lab2

Кога се користат регуларни изрази во комбинација со `grep`, најдобро е регуларниот израз секогаш да се загради со `' '`, што ќе го спречи интерпретерот да го процесира изразот пред да го предаде на `grep`.

Со:

```
student@os:~$ grep -n '^\$' money
```

ќе се излистаат сите линии во датотеката `money` кои започнуваат со `$` заедно со нивниот реден број.

```
student@os:~$ ls -l | grep '^d.*'
```

ќе ги излиста детално сите именици во тековниот именик.

Со:

```
student@os:~$ grep 'main' *.c
```

ќе се излистаат сите линии од сите датотеки со наставка `.c` во тековниот именик кои содржат низа `main`, а додека пак со

```
student@os:~$ grep '<main>' *.c
```

ќе се излистаат сите линии од сите датотеки со наставка `.c` во тековниот именик кои го содржат зборот `main`.

Со:

```
student@os:~$ ls -l | grep '\.txt$' | cat > poraka
```

ќе се даде детален извештај на сите датотеки кои ја имаат наставката `.txt` и притоа излезот ќе се пренасочи во датотеката `poraka`.

Командата `cut`

Командата `cut` врши издвојување колони од податоци. Односно командата користи IFS (Input Field Separator) кој препознава по што да се врши одделувањето. Се користи заедно со следниве опции:

-d ' : ' - делимитер

-f x (каде x е цел број) - врши одделување на x-та колона

-c x-y (каде x и y се цели броеви) - врши одделување на карактерите од x до y во податокот

Пример:

```
student@os:~$ ls -l Lab2 | grep '^-' | tr -s ' ' | cut -f 5 -d ' '
```

Во овој пример се врши листање на сите датотеки од директориумот Lab2, а потоа со наредбата `tr -s ' '` се врши спојување на повеќе празни места во едно затоа што може да се случи да се јават повеќе празни места помеѓу колоните при листањето со `ls -l` и притоа наредбата `cut` да не ни ги даде посакуваните резултати. Овој излез со помош на цевка се пренасочува на наредбата `cut` која што ќе ја издвои 5-тата колона (односно големината на датотеките) од останатите колони кои се одделени со по едно празно место.

Командата `awk`

Кога ја користиме `awk` во комбинација со `print`, тоа значи да се испечатат само специфицирани полиња. Се користи на следниов начин:

```
$awk '{print $бр_колона, $ бр_колона, ... , $ бр_колона;}'
```

Пример:

```
student@os:~$ ls -l | grep '^-' | awk '{print $1, $5, $8;}'
```

Ќе се испечатат на екран за секоја датотека од вашиот тековен именик: привилегиите, нејзината големина и името.

Операторот `grave` ``

Во случаи кога е потребно резултатот од извршување на команда да се смести во системска променлива може да се користи *grave* операторот (``` команда ``` каде наводниците се оние под копчето Escape кај `~`). Тој предизвикува извршување на командата и пренасочување на стандардниот излез во променливата. Во следниот пример, содржината на тековниот именик се сместува во променливата листа:

```
student@os:~$ lista=`ls`  
student@os:~$ echo "Sodrzinata na tekovniot imenik: $lista"
```

додека со:

```
student@os:~$ echo "Ima `ls $HOME | wc -l` datoteki vo $HOME" Ima  
51 datoteki vo /home/student
```

се прикажува бројот на датотеки во именикот \$HOME. (Напомена уште еднаш: да се внимава на наводниците кај grave операторот!)

За на час

1. Запишете го излезот од командите во датотеката `zad1.txt`:

```
echo "Najaveni se:" who echo > zad1.txt
```

```
echo "Vкупno:" who | wc -l >> zad1.txt
```

2. Пронајди ги сите датотеки во тековниот именик со екстензија `.txt` и резултатот запишете ги во датотекат `zad2.txt`.

```
ls *.txt > zad2.txt
```

3. Избројте колку датотеки и подименици (заедно) имате во тековниот именик и излезот запишете го во датотеката `zad3.txt`.

```
ls | wc -l > zad3.txt
```

4. Креирај датотека `zad4.txt` со следнава содржина:

```
Zz .. ss
```

```
Ll . k L ... io
```

```
echo -e "Zz .. ss\nLl . k L ... io" >zad4.txt
```

5. Излистајте ги датотеките кои во својата содржина го содржат карактерот `.` (точка) и резултатот запишете го во датотеката `zad5.txt`.

```
grep '\.' *.txt > zad5.txt
```

6. Избројте колку корисници чии индекси започнуваат на `11` се најавени моментално на системот и излезот запишете го во датотеката `zad6.txt`.

```
who | grep '^11' | wc -l > zad6.txt
```

7. Избројте колку именици има во вашиот домашен именик и излезот запишете го во датотеката `zad7.txt`.

```
ls $HOME | wc -l > zad7.txt
```

8. Да се испишат само имињата на сите датотеки (без именици) во тековниот именик и излезот запишете го во датотеката `zad8.txt`. (**Упатство:** Користете цевки, `grep` и `awk`. Да се игнорираат имениците и датотеките кои содржат празно место во нивното име).

```
ls -l | grep '\.' | awk '{print $9}' | grep -v '\s'
```

9. Со користење на `sed` командата, да се испишат само имињата на сите именици (без датотеки) во тековниот именик, а потоа да се сортираат по азбучен редослед и резултатот да се пренасочи во датотека `zad9.txt`.

```
ls | grep -v '\.txt$' | sed -e "s/[w]/u/g" | sort -f
```

10. Да се испишат сите датотеки и именици кои имаат привилегија за запишување од страна на корисникот и резултатот да се пренасочи во датотека `zad10.txt`
`ls -l | grep '^..w' | awk '{print $9}' > zad10.txt`
11. Да се излистаат сите најавени корисници чие корисничко име завршува на 5 по азбучен редослед и времето кога тие се најавиле и резултатот да се пренасочи во датотека `zad11.txt`
`who | tr -s ' ' | awk '{print $1,$3,$4}' | grep '3\b ' | sort -f > zad11.txt`
12. Да се направи копирање на сите датотеки кои започнуваат на `zad` и се со екстензија `.txt` во именикот `zadaci`. Да се креира именикот `zadaci` во домашниот именик.
`cp zad*.txt zadaci/`

Задачи за вежбање

Напишете регуларен израз кој :

- Има барем едно `i` по што следат произволен број на броеви: `_i[0-9]*` _____
- Било кои 5 карактери: `_.\{5\}` _____
- Зборови од 3 букви првата е `s` крајната е `t` а помеѓу може да е било која самогласка: `__\bc[aeiou]t\b` _____
- Зборови од 3 букви првата е `s` крајната е `t` а измеѓу може да е било која согласка: `__\bc[^aeiou]t\b` _____
- Зборови од 5 букви каде крајната е `s` а содржат и точка: _____
- Зборови што почнуваат со `s` а завршува со `t` а измеѓу има 5 до 10 карактери "а": `__\bca\{5,10\}t\b` _____
- Зборови што почнуваат со `ii` а завршува со `student` а измеѓу има 4 до 6 цифри: `__\bii[0-9]\{4,6\}student\b` _____
- Зборови што почнуваат на "р" и завршуваат на самогласка: `__\bp.*[aeiou]\b` _____
- Зборови што почнуваат со точка ".": `__\b\.` _____
- Зборови што започнуваат со `a`, а завршуваат со `z` а измеѓу има неопределно број букви: `__\ba.*z\b` _____