

LD3. Serverių žurnalų analizės scenarijai

Aprašymas

Darbiui naudosime scenarijų kalbą *Perl*. Trumpas *Perl* sintaksės aprašymas:

Operacijos su kintamaisiais:

```
$x = 5 + 9;
$x = 30 - 4;
$x = 3 * 7;
$x = 6 / 2;
$x = 2 ** 8; #kelimas laipsniu
$x = 3 % 2;
$y = ++$x;
$y = $x++;
$y = --$x;
$y = $x--;
$x = $y;
$x += $y;
$x -= $y;
$x .= $y; # Prijungti $y prie $x
```

Sąlygos sakinyis if:

```
if($x == $y){
    print "x lygus y";
}
else{
    print "x nelygus y";
}

if ($vardas eq "Darius") { #string
    print "Jūsų vardas yra 'Darius'\n";
}
```

Ciklas for:

```
for $i (1, 2, 3, 4, 5) {
    print "$i\n";
}
for($i = 1; $i <= 10; $i++){
    print "for ciklas#$i\n";
}
```

Ciklas while:

```
$x = 0;
while($x < 5){
    print "$x\n";
    $x++;
}
$x = 0;
do{
    print "$x\n";
    $x++;
}while($x < 5);
```

Ciklas until:

```
$i = 0;
until ($i++ > 5) {
    print "i lygus $i\n";
}
```

Masyvai:

```
@str_array = ("Ac","Darius","Stalas");
@int_array = (5,7,9,10);
$str_array[1] = 5;
push(@mas,10); #papildyti į pab.
$last = pop(@mas); #pašalinti iš pab
unshift(@mas,0);
#papildyti 0 į @mas prad.
$start = shift(@mas);
# papildyti 0 į prad.
print $#mas; # Elementų kiekis-1
```

Masyvų sujungimas:

```
@mas = (@mas1, @mas2, "Gruodis");
```

Masyvas kaip eilutė (tarp elementų tarpai):

```
$str = join(' ',@mas);
```

Argumentų perdavimas:

```
foreach $arg (@ARGV) {
    print $arg, "\n";
}
print "@ARGV[0]\n"
```

Argumentų kiekis:

```
$kiekis = $#ARGV + 1;
```

Failų skaitymas ir rašymas. Failų atidarymui naudojama funkcija `open`, kurios pirmasis argumentas – failo nuoroda į failą (handle), antrasis – failo pavadinimas:

```
open (FAILAS, "failas.txt");
open (FAILAS, "failas.txt") or
die ("Failas neatidarytas!");
```

Failo uždarymas:

```
close (FAILAS);
```

Visų failo eilučių nuskaitymas į masyvą:

```
@eilutės = <FAILAS>;
```

Failo atidarymas rašymui ir rašymas:

```
open (FAILAS, ">>failas.txt") or die
"Nepakankamos privilegijos";
open (FAILAS, ">failas.txt") or die
"Nepakankamos privilegijos";
```

```
print FAILAS "Elementų kiekis:
$kiekis";
close (FAILAS);
```

Atvaizdžiai:

```
%svetaines = (
    "ktu.lt" => "KTU",
    "delfi.lt" => "Delfi"
);
```

Atvaizdžio elementai pasiekiami naudojant raktą:
`print $svetaines{"ktu.lt"}`

Atvaizdžio papildymas nauju elementu:
`$svetaines{"lrytas.lt"} = "Lrytas";`

Patikrinimas ar atvaizdyje egzistuoja elementas su duotu raktu (naudojama funkcija `exists`):
`if (exists $svetaines{"lrytas.lt"}) {
 print $svetaines{"lrytas.lt"};
}`

Elementų pašalinimas. Vieno elemento:
`delete $svetaines{"lrytas.lt"};`

Visų elementų:
`%svetaines = ();`

Raktai grąžinami į masyvą:

```
@svetainiu_adresai = keys %svetaines;
```

Pereiti per visus atvaizdžio elementus:

```
foreach $svetaine (keys %svetaines) {  
    print "$svetaine atitinka  
    $svetaines{$svetaine}\n";  
}
```

Atvaizdis gali būti priskiriamas masyvui (ar atvirkščiai) arba kitam atvaizdžiui:

```
@svetaines = %svetaines;  
%svetaines = @svetaines;  
%adresai = %svetaines;
```

Įdiegtos funkcijos:

Grąžina simbolių kiekį string eilutėje.
`length("Šiandien yra pirmadienis");`

Grąžina eilutės dalį. Pirmas argumentas – eilutė, antras – postūmis nuo pradžių, trečias – elementų kiekis nuo postūmio.
`substr("A long string", 3);
substr("A long string", 0, 6);`

Suskaido eilutę į masyvą:
`@mas = split(" ", $str);
@mas = split(/\d+/, $str);
skyriklis – simbolis arba regexp
@mas = split(' ', $data, 3)
skaldo pirmus 3 kartus.`

Gražina masyvą su duomenimis apie failą:
`($dev, $ino, $mode, $nlink, $uid,
$gid, $rdev, $size, $atime, $mtime,
$ctime, $blksize, $blocks) =
stat("eof.dat");`

Spausdinimas:
`printf("January = \${%8.2f}\n",
$januaryCost);`

Darbas su direktorijomis:

```
opendir(DIR, "/home")  
    or die "Nera tokios direktorijos";  
  
while ($file = readdir(DIR)) {  
    print " $file\n";  
}  
closedir(DIR);
```

```

Surikiuoti direktorijos failus:
foreach $file (sort readdir(DIR)) {
    print " $file\n";
}

```

Web serverio prieigos žurnalai (Web Access Log). Juose saugoma įvairi statistika apie iš web serverio parsisiunčiamus pasiekiamus web puslapius. Apache web serveris šiame žurnale kaupia statistiką tokiu formatu:

64.242.88.10 - - [07/Mar/2004:19:19:19 -0800] "GET /mailman/listinfo/mlc HTTP/1.1" 200 6142

Kur:

64.242.88.10	Kliento IP adresas.
"_"	Kliento RFC 1413 identitetas (šiuo atveju nėra).
"_"	Vartotojo id.
[07/Mar/2004:19:19:19 -0800]	Laikas kada serveris baigai apdoroti užklausą.
"GET /mailman/listinfo/mlc HTTP/1.1"	Užklauskos eilutė iš kliento (kabutėse).
200	Būsenos kodas, kurį serveris atgal siunčia klientui. Galimi kodai: 200 - OK 206 - Partial Content 301 - Moved Permanently 302 - Found 304 - Not Modified 401 - Unauthorised (password required) 403 - Forbidden 404 - Not Found
6142	Klientui grąžinamo objekto dydis baitais

Tokį failą su labai įvairiu formatavimu Perl'e patogiu nuskaityti naudojantis reguliariosiomis išraiškomis, kurios apskliaudžiamos operatoriais //:

```

if ($eilute ~ /reguliarioji_išraiška/){
    print $eilute;
}

```

Gali būti naudojami operatoriai:

- (m//) - atitikimas.
- (s//) – pakeitimas (Taip kaip sed).

Taip pat specialieji simboliai: \d (skaitmuo), \s (tarpas), \w (žodis). \D, \S, \W yra \d\s\w neiginiai. Pavyzdžiui, minėto žurnalo failo reguliarioji išraiška ir failo nuskaitymas galėtų atrodyti taip:

64.242.88.10 - - [07/Mar/2004:19:19:19 -0800] "GET /mailman/listinfo/mlc HTTP/1.1" 200 6142

```

/^(\\s+) (\\s+) (\\s+) \\[(.+)\\] \\["(.+)\\"] (\\s+) (\\s+)/

```

```

while(<FAILAS>){ #skaitymos failo eilutės
    chomp;      #nuimami tušti simboliai pradžioje ir pabaigoje
}

```

```
s/\s+/ /go; ## keli tarpai apjungiami į vieną tarpą

($ip, $rfc1413, $userid, $laikas, $httpUzklausa, $busenosKodas, $baitai) =
/^(\\S+) (\\S+) (\\S+) \\[(.+)\\] \\\"(.+)\\\" (\\S+) (\\S+)/o;
}
```

Užduotys

Serverio adresas: **158.129.0.113**. Prisijungimas per *putty* arba *Secure Shell Client*. Vartotojo vardas –*varpava*. Scenarijaus failo pradžioje reikia nurodyti kelią iki perl interpretatoriaus:

```
#!/usr/bin/perl -w
```

Užduotis1. Parašykite scenarijų *uzd9.pl*, kuris suskaičiuotų kiek žurnale */home/stud/stud/access_log* saugoma įrašų su argumente nurodomu IP adresu ir kiek baitų perduota šiuo IP adresu. Failo eilutės skaidymui panaudokite perl funkcijas *split* ir *length*. Rezultatus išveskite į failą *rez*.

Užduotis2. Modifikuojant scenarijų *uzd9.pl* parašykite scenarijų *uzd10.pl*, kuris papildomai tenkintų tokias sąlygas:

- Jei argumentas nėra nurodomas, turi būti atspausdinamas visų skirtingų žurnalo IP adresų sąrašas, o prie kiekvieno jų atspausdinama kiek žurnale yra su juo susijusių įrašų, o taip pat kiek jie užima sėkmingai perduotų baitų (tikrinimas ar būsenos kodas yra 200).
- Tai atlikite naudodami perl atvaizdį (angl. Dictionary).

Užduotis3. Parašykite scenarijų *uzd11.pl*, kuris naudojantis žurnalą */home/stud/stud/access_log* išspausdintų visus skirtingus IP adresus, kuriais vartotojas kreipiasi į serverį, kada jam grąžinamas atsakymas, kad tokio puslapio nėra (klaidos kodas 404).

Užduotis4 (papildomai). Parašykite scenarijų *uzd12.pl*, kuris suskaičiuotų pirmu argumentu nurodomame kataloge esančių katalogų užimamą vietą baitais, ir išspausdintų tik tų katalogų sąrašą, kurių užimama vieta viršija antru argumentu nurodomą dydį. Šalia kiekvieno spausdinamo katalogo išspausdinkite penkis didžiausius jame esamus failus. Paprastumo dėlei laikykite, kad kiekvienoje direktorijoje nėra kitų direktorijų.