

Ειδικά Θέματα Βιοτληροφορικής

Διάλεξη 5η :

Perl (3) : Καθιερωμένη είσοδος: αλλαγή του τρόπου ανάγνωσης, Είσοδος/έξοδος από αρχεία,
Τύποι μεταβλητών : hash arrays, Άσκηση 3η.

Ανάγνωση από την καθιερωμένη είσοδο

Έχουμε ήδη δει πως η χρήση της έκφρασης `<STDIN>` επιτρέπει την ανάγνωση μιας ολόκληρης γραμμής (μέχρι και το τελικό `\n`) από την καθιερωμένη είσοδο. Εάν όμως επιθυμούμε να διαβάζουμε από την είσοδο όχι ανά γραμμή, αλλά ανά χαρακτήρα, ή παράγραφο ;
Η λύση της perl είναι ο επανακαθορισμός μίας εσωτερικής μεταβλητής, της `$/`, η τιμή της οποίας καθορίζει πόσα bytes μπορούν να διαβαστούν από την είσοδο ή ποιοι χαρακτήρες σηματοδοτούν το τέλος γραμμής.

Ανάγνωση από την καθιερωμένη είσοδο

To νόημα του \$/

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

$/ = \1;
while ( $line = <STDIN> )
{
    print "$line\n";
}
```

Ανάγνωση από την καθιερωμένη είσοδο

To vόημα του \$/

```
# ./test.pl
This is a test
T
h
i
s
i
s
a
t
e
s
t
```

Ανάγνωση από την καθιερωμένη είσοδο

To νόημα του \$/

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

$/ = 1;
while ( $line = <STDIN> )
{
    print "$line\n";
}
```

Ανάγνωση από την καθιερωμένη είσοδο

To νόημα του \$/

```
# ./test.pl
This is 1 a test but what 1 isn't ? 1
This is 1
a test but what 1
isn't ? 1
```

Ανάγνωση από την καθιερωμένη είσοδο

To νόημα του \$/

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

$/ = "\n\n";
while ( $line = <STDIN> )
{
    print "$line\n";
}
```

Ανάγνωση από την καθιερωμένη είσοδο

To νόημα του \$/

```
# ./test.pl
```

```
This is the first  
paragraph.
```

```
This is the first  
paragraph.
```

```
This is the second  
(and longer)  
paragraph.
```

```
This is the second  
(and longer)  
paragraph.
```

Είσοδος/έξοδος από αρχεία

Η χρήση αρχείων για είσοδο/έξοδο γίνεται σε τρία βήματα. Το πρώτο βήμα είναι η δημιουργία μίας μεταβλητής (file handle) μέσω της οποίας γίνονται όλες οι αναφορές στο αρχείο. Αυτό επιτυγχάνεται με την εντολή open. Το δεύτερο βήμα είναι η χρήση των file handles για είσοδο/έξοδο, παρόμοια με τη χρήση της έκφρασης <STDIN> που έχουμε ήδη δει. Το τελευταίο βήμα είναι το κλείσιμο των αρχείων με την εντολή close. Με τη μορφή παραδειγμάτων:

Είσοδος από αρχεία

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

# Δημιουργία του file handle "MYFILE" για
# ανάγνωση από το αρχείο data.in στον
# τρέχοντα κατάλογο.
open ( MYFILE, "data.in" );

# Χρήση του MYFILE για ανάγνωση
while ( $line = <MYFILE> )
{
    print "$line";
}

# Κλείσιμο του MYFILE
close ( MYFILE );
```

Είσοδος από αρχεία

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

open ( FILE1, "A.in" ) || die "Can not open A.in\n";
open ( FILE2, "B.in" ) || die "Can not open B.in\n";

while ( $line1 = <FILE1> , $line2 = <FILE2> )
{
    chomp( $line1 );
    print "$line1 $line2";
}

close ( FILE2 );
close ( FILE1 );
```

Είσοδος από αρχεία

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

open ( FILE1, "A.in" ) or die "Can not open A.in\n";
open ( FILE2, "B.in" ) or die "Can not open B.in\n";

while ( $line1 = <FILE1> , $line2 = <FILE2> )
{
    chomp( $line1 );
    print "$line1 $line2";
}

close ( FILE2 );
close ( FILE1 );
```

Είσοδος από αρχεία

```
# cat A.in
1
2
3
7
# cat B.in
4
5
6
# ./test.pl
1 4
2 5
3 6
```

Παράμετροι γραμμής εντολής

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

open ( FILE1, $ARGV[0] ) or die "Can not open $ARGV[0]\n";
open ( FILE2, $ARGV[1] ) or die "Can not open $ARGV[1]\n";

while ( $line1 = <FILE1> , $line2 = <FILE2> )
{
    chomp( $line1 );
    print "$line1 $line2";
}

close ( FILE2 );
close ( FILE1 );
```

Παράμετροι γραμμής εντολής

```
# ./test.pl A.in B.in
1 4
2 5
3 6
#
#
# ./test.pl
Use of uninitialized value in open at ./tt.pl line 5.
Use of uninitialized value in concatenation (.) at
./tt.pl line 5.
Can not open
#
```

Παράμετροι γραμμής εντολής

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

( @ARGV == 2 ) or die "Usage: test.pl file1 file2\n";

open ( FILE1, $ARGV[0] ) or die "Can not open $ARGV[0]\n";
open ( FILE2, $ARGV[1] ) or die "Can not open $ARGV[1]\n";

while ( $line1 = <FILE1> , $line2 = <FILE2> )
{
    chomp( $line1 );
    print "$line1 $line2";
}

close ( FILE2 );
close ( FILE1 );
```

Παράμετροι γραμμής εντολής

```
# ./test.pl A.in B.in
1 4
2 5
3 6
#
#
# ./test.pl
Usage: test.pl file1 file2
#
```

Έξοδος σε αρχεία

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

( @ARGV == 2 ) or die "Usage: test.pl file1 file2\n";

open ( IN_1, $ARGV[0] ) or die "Can not open $ARGV[0]\n";
open ( IN_2, $ARGV[1] ) or die "Can not open $ARGV[1]\n";

open ( OUT, ">output.dat" ) or die "Can not open output.dat\n";

while ( $line1 = <IN_1> , $line2 = <IN_2> )
{
    chomp( $line1 );
    print OUT "$line1 $line2";
}

close ( OUT );
close ( IN_2 );
close ( IN_1 );
```

Έξοδος σε αρχεία

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

( @ARGV == 3 ) or die "Usage: test.pl file1 file2 outfile\n";

open ( IN_1,    $ARGV[0]  ) or die "Can not open $ARGV[0] for reading\n";
open ( IN_2,    $ARGV[1]  ) or die "Can not open $ARGV[1] for reading\n";
open ( OUT, ">$ARGV[2]" ) or die "Can not open $ARGV[2] for writing\n";

while ( $line1 = <IN_1> , $line2 = <IN_2> )
{
    chomp( $line1 );
    print OUT "$line1 $line2";
}

close ( OUT );
close ( IN_2 );
close ( IN_1 );
```

Έξοδος σε αρχεία

```
# ./test.pl
Usage: test.pl file1 file2 outfile
# cat A.in
1
2
3
7
# cat B.in
4
5
6
# ./test.pl A.in B.in out.dat
# cat out.dat
1 4
2 5
3 6
```

Προσθήκη εξόδου σε αρχεία

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

( @ARGV == 3 ) or die "Usage: test.pl file1 file2 outfile\n";

open ( IN_1,    $ARGV[0]  ) or die "Can not open $ARGV[0] for reading\n";
open ( IN_2,    $ARGV[1]  ) or die "Can not open $ARGV[1] for reading\n";
open ( OUT, ">>$ARGV[2]" ) or die "Can not open $ARGV[2] for appending\n";

while ( $line1 = <IN_1> , $line2 = <IN_2> )
{
    chomp( $line1 );
    print OUT "$line1 $line2";
}

close ( OUT );
close ( IN_2 );
close ( IN_1 );
```

Hash arrays

Στους πίνακες που έχετε δει μέχρι τώρα, η αναφορά σε κάποιο στοιχείο τους γίνεται μέσω ενός αριθμητικού δείκτη, π.χ. η έκφραση \$array[1] αντιστοιχεί στο δεύτερο στοιχείο του @array.

Εάν θέλαμε να αντιστοιχίσουμε ένα στοιχείο του πίνακα σε κάποια λέξη-κλειδί (π.χ. το γράμμα "W" στο μοριακό βάρος της τρυπτοφάνης) θα χρειαζόμασταν ενδιάμεσους πίνακες (δείκτες σε δείκτες). Η perl απλοποιεί κατά πολύ τα πράγματα μέσω της χρήσης των hashes τα οποία είναι πίνακες των οποίων οι δείκτες είναι λέξεις κλειδιά. Τα ονόματα των hashes αρχίζουν με το σύμβολο '%'. Με τη μορφή παραδειγμάτων:

Hash arrays

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

%MWs = ( "A" => 71.08,
          "C" => 103.14,
          "D" => 115.09 );

print $MWs{ "A" }, "\n";
print "$MWs{ \"A\" }\n";

$MWs{ "E" } = 129.12;
print $MWs{ "A" } + $MWs{ "E" }, "\n";

if ( defined( $MWs{ "P" } ) )
{
    print $MWs{ "P" }, "\n";
}
else
{
    print "P key is has not been defined\n";
}
```

Hash arrays

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

%MWs = ( "A" => 71.08,
          "C" => 103.14,
          "D" => 115.09 );

print $MWs{ "A" }, "\n";
print "$MWs{ \"A\" }\n";

$MWs{ "E" } = 129.12;
print $MWs{ "A" } + $MWs{ "E" }, "\n";

if ( defined( $MWs{ "P" } ) )
{
    print $MWs{ "P" }, "\n";
}
else
{
    print "P key is has not been defined\n";
}
```

Hash arrays

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

%MWs = ( "A" => 71.08,
          "C" => 103.14,
          "D" => 115.09 );

print $MWs{ "A" }, "\n";
print "$MWs{ \"A\" }\n";

$MWs{ "E" } = 129.12;
print $MWs{ "A" } + $MWs{ "E" }, "\n";

if ( defined( $MWs{ "P" } ) )
{
    print $MWs{ "P" }, "\n";
}
else
{
    print "P key is has not been defined\n";
}
```

Hash arrays

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

%MWs = ( "A" => 71.08,
          "C" => 103.14,
          "D" => 115.09 );

print $MWs{ "A" }, "\n";
print "$MWs{ \"A\" }\n";

$MWs{ "E" } = 129.12;
print $MWs{ "A" } + $MWs{ "E" }, "\n";

if ( defined( $MWs{ "P" } ) )
{
    print $MWs{ "P" }, "\n";
}
else
{
    print "P key is has not been defined\n";
}
```

Hash arrays

```
# ./test.pl  
71.08  
71.08  
200.2  
P key is has not been defined
```

Hash arrays

Χρήση των lc και uc.

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

%MWs = ( "A" => "Capital A",
          "a" => "Lower    a" ) ;

$char_cap = "A";
$char_low = "a";

print $MWs{ $char_cap }, "\n";
print $MWs{ lc ( $char_cap ) }, "\n";

print $MWs{ $char_low }, "\n";
print $MWs{ uc( $char_low ) }, "\n";
```

Hash arrays

Χρήση των lc και uc.

```
# ./test.pl
Capital A
Lower   a
Lower   a
Capital A
```

ΑΣΚΗΣΗ

Γράψτε το πρόγραμμα σε perl το οποίο θα διαβάζει από την καθιερωμένη είσοδο μία πρωτεϊνική αλληλουχία (μέχρι να ανιχνευθεί τέλος εισόδου) και θα τυπώνει στην καθιερωμένη έξοδο το συνολικό μοριακό βάρος της αλληλουχίας που διάβασε. Βεβαιωθείτε ότι το πρόγραμμα σας δίνει σωστά αποτελέσματα, και στη συνέχεια στείλτε το στον διδάσκοντα:

```
mail -s "My name, Pract 3" glykos@aspera.cluster.mbg.gr < myprog.pl
```

ΑΣΚΗΣΗ

A	71.1	C	103.1
D	115.0	E	129.1
F	147.1	G	57.1
H	137.1	I	113.1
K	128.1	L	113.1
M	131.1	N	114.1
P	97.1	Q	128.1
R	156.1	S	87.1
T	101.1	V	99.1
W	186.2	Y	163.1