

PERL



Δήμος Παύλου  
Δημήτρης Κουζαπάς

# Perl

- Η Perl είναι μια διερμηνευομένη δυναμική γλώσσα προγραμματισμού.
- Συνδυάζει χαρακτηριστικά από διάφορες γλώσσες. (C/C++, Awk, Lisp κλπ)
- Διαδόθηκε λόγω της μεγάλης δυνατότητας της στην επεξεργασία συμβολοσειρών.

# Ιστορία

- Αναπτύχθηκε από τον Larry Wall.
- Η πρώτη έκδοση κυκλοφόρησε το 1987
- Η δεύτερη έκδοση τον επόμενο χρόνο με βελτιωμένη μηχανή κανονικών εκφράσεων
- Η τρίτη έκδοση κυκλοφόρησε το 1989 με υποστήριξη για δυαδικά δεδομένα
- Η τέταρτη έκδοση έγινε το 1993

# Ιστορία

- Η σημερινή έκδοσή της Perl είναι η πέμπτη έκδοση που κυκλοφόρησε το 1994
- Ο διερμηνέας της Perl ξαναγράφηκε
- Η γλώσσα υποστηρίζει αντικειμενοστραφή προγραμματισμό
- Υπάρχει δυνατότητα προσθήκης βιβλιοθηκών (modules).
- Comprehensive Perl Archive Network.  
Παγκόσμια βιβλιοθήκη με modules.

# Δυνατότητες – Περιορισμοί

- Η Perl συνδυάζει χαρακτηριστικά από διάφορες γλώσσες.
- Έχει παρόμοια σύνταξη με τη C.
- Η ύπαρξη δομών όπως lists, hashes και οι κανονικές εκφράσεις επιτρέπουν την εύκολη επεξεργασία συμβολοσειρών
- Υποστηρίζει αντικειμενοστραφή προγραμματισμό.
- Portability. Λόγω του ότι η γλώσσα είναι διερμηνευόμενη.

# Δυνατότητες - Περιορισμοί

- Η Perl τρέχει με διερμηνέα και αυτό δεν επιτρέπει την ανάπτυξη αποδοτικών προγραμμάτων.
- Ο προγραμματιστής δεν έχει έλεγχο της επεξεργασίας δομών δεδομένων όπως οι συμβολοσειρές και hashes.
- Υπάρχουν πολλά γνωρίσματα σε ψηλό επίπεδο.

# Εγκατάσταση

- Ανάκτηση source code ή binary
  - <http://www.perl.com/download.csp>
- Διαθέσιμα binaries για Unix/Linux, Windows και Macintosh
- Στις περισσότερες πλατφόρμες Linux γίνεται by default εγκατάσταση του εργαλείου.

# Hello World!

- Κώδικας για Hello World!

```
$ cat -n hello.pl
  1 print "Hello World!\n";
$ perl hello.pl
Hello World!
```

```
$ cat -n hello.pl
  1 #!/usr/bin/perl
  2 print "Hello World!\n";
$ chmod +x hello.pl
$ ./hello.pl
Hello World!
```



# Βασικές γνώσεις

- Δύο τρόποι εκτέλεσης παρόμοιοι με όλες τις άλλες scripting γλώσσες
- 1<sup>ος</sup> τρόπος:
  - `$ perl program.pl`
- 2<sup>ος</sup> τρόπος:
  - Στην πρώτη γραμμή του κώδικα χρησιμοποιείται το `#!/usr/bin/perl` (ή το απόλυτο μονοπάτι για τον interpreter)
  - `$ chmod +x program.pl`
  - `$ ./program.pl`

# Βασικές γνώσεις (Συνέχεια)

- Δυνατότητα εκτέλεσης αντικατάστασης κειμένου από το command line:
  - `$ perl -p -i.bak -e "s/from/to/options;" Files`
    - Ψάχνει σε όλα τα files για την έκφραση "from" και την αντικαθιστά με την έκφραση "to".
    - Η παράμετρος -i δημιουργεί backup των πρωτότυπων αρχείων με το extension .bak. Αν δεν δοθεί extension τότε τα κάνει overwrite.
    - Αντί για το σύμβολο '/' για διαχωριστικό των εκφράσεων, μπορεί να χρησιμοποιηθεί οποιοδήποτε σύμβολο δεν υπάρχει στις εκφράσεις "from" και "to".

**sed????**

# Βασικές γνώσεις (Συνέχεια)

- Τέσσερις τύποι μεταβλητών
  - (\$) Scalars:
    - Απλά strings ή αριθμητικές τιμές, δείκτες ή typeglobs.
  - (@) Arrays/Lists:
    - Μονοδιάστατος πίνακας με 0 ή περισσότερα scalars. Μπορούμε να δούμε την τιμή κάποιου στοιχείου με χρήση των []. Το indexing γίνεται όπως στη C/C++ ξεκινώντας από το 0.

# Βασικές γνώσεις (Συνέχεια)

- (%) Hash:
  - Μια λίστα από ζεύγη (κλειδί, τιμή). Με αυτή τη δομή μπορεί να γίνει αποδοτικά η αναζήτηση κάποιου συγκεκριμένου κλειδιού.
- (\) Reference:
  - Ο δείκτης αναφέρεται σε κάποια άλλη τιμή όπως τους δείκτες της C/C++.

# Βασικές γνώσεις (Συνέχεια)

- Εντολές ελέγχου ροής εκτέλεσης
  - Σε αντιπαράβολή με τη C:

	C	Perl (braces required)
same	<code>if () { ... }</code>	<code>if () { ... }</code>
diff	<code>} else if () { ... }</code>	<code>} elsif () { ... }</code>
same	<code>while () { ... }</code>	<code>while () { ... }</code>
diff	<code>do while ();</code>	<code>do while (); (See below)</code>
same	<code>for (aaa;bbb;ccc) { ... }</code>	<code>for (aaa;bbb;ccc) { ... }</code>
diff	N/A	<code>foreach \$var (@array) { ... }</code>
diff	<code>break</code>	<code>last</code>
diff	<code>continue</code>	<code>next</code>
similar	<code>0 is FALSE</code>	<code>0, "0", and "" is FALSE</code>
similar	<code>≠ 0 is TRUE</code>	<code>anything not false is TRUE</code>

Σημείωση: Τα curly brackets δεν είναι προαιρετικά στην Perl όπως σε ορισμένες περιπτώσεις στη C/C++

# Βασικές γνώσεις (Συνέχεια)

- Τελεστές

Description	string op	numeric op
equality	eq	==
inequality	ne	!=
ternary compare	cmp	<=>
concatenation	. (a dot)	N/A
arithmetic	N/A	+, -, *, /
relational	lt, le, gt, ge	<, <=, >, >=
ANSI C ops		

# Επεξεργασία αρχείου με κείμενο

- Εξαγωγή τιμών από αρχείο που περιέχει αποτέλεσμα κάποιας προσομοίωσης
  - Χρήση file handles
  - Χρήση string functions
- Το script παίρνει ως παραμέτρους από το command line το όνομα του αρχείου με τα αποτελέσματα και το ένα όνομα αρχείου όπου θα αποθηκευτούν οι τιμές.
- Στο τέλος της εκτέλεσης οι τιμές τυπώνονται στην οθόνη και το αρχείο.

# Επεξεργασία αρχείου με κείμενο

```
#!/usr/bin/perl
# Extract several values from the input files which contain
# the results of a binary execution on the SimpleScalar simulator

# Modules which are going to be used
use English;      # use nice English for ugly punctuation variables
use FileHandle;  # supply object methods for filehandles
use Carp;         # warn of errors (from perspective of caller)
use strict;      # Install all three strictures (vars, refs, subs)

# Variable Declarations - Global

# Scalars
my($numberOfKeyWords);
my($inputFileName);

# Arrays
my(@keyWords);
my(@tempKeywordVariables); #Used as a temporary array to hold the values read
```



# Επεξεργασία αρχείου με κείμενο

```
# Main program function
sub main()
{
    # Check for correct number of arguments
    if( @ARGV != 1) {
        print "Please give the file with the results\n";
        exit 0;
    }

    # The keywords that are going to be found
    $keyWords[0] = "sim_IPC";
    $keyWords[1] = "i11.miss_rate";
    $keyWords[2] = "d11.miss_rate";

    # Set the number of keywords
    $numberOfKeyWords = @keyWords;

    # Set the inputFileName
    $inputFileName = $ARGV[0];

    handleFile($inputFileName);

    printResults();
}
}
```

# Επεξεργασία αρχείου με κείμενο

```
# Handle file
sub handleFile($)
{
    # Copy the parameter array in the scalar
    my($infile) = @_;

    # open the file
    my($IN) = new FileHandle "$infile";

    if(!defined($IN)) {
        print "Can't open file $infile: $!\n";
    }

    processFile($IN, "$infile");
    $IN->close();
}
```

# Επεξεργασία αρχείου με κείμενο

```
# Process the input file to extract the values of the keywords
sub processFile($$)
{
    # Temporary variables
    my($tt1);
    my($tt2);
    my($cv01);

    my($IN, $filename) = @_;
    my($line, $from) = ("", "");

    print "Processing: ".$filename."\n";
}
```

# Επεξεργασία αρχείου με κείμενο

```
while ( defined($line=<$IN>)) {
    # remove the white spaces of the line
    chomp($line);

    $cv01 = 0;

    # For all the keywords do
    foreach (@keyWords) {
        # check if the line contains the keywords
        if($line =~ /$_/) {
            #Remove the search keyword from the line and everything before it
            $tt1=rindex($line,$_);
            $tt2=substr($line,($tt1+length($_)),length($line));
            $line=$tt2;

            #Match the regular expression
            $line =~ /(:)*^?\ *(\d+(\.\d*)?|\.\d+)([eE][+-]?\d+)*([\s|\w|\W]*)?$/;

            #The value read is stored in the $tempKeywordVariables array
            $tempKeywordVariables[$cv01]=(defined($4) ? ($2." ".$4):($2));
        }
        $cv01 = $cv01 + 1;
    }
}
```



# Επεξεργασία αρχείου με κείμενο

```
# Print results
sub printResults($)
{
    my($outFileName) = @_;
    my($tmpOut) = new FileHandle "> $outFileName";

    if(!defined($tmpOut)) {
        print "Can't open file $outFileName: $!\n";
    }

    #Local variable
    my($cv01);

    #For all keywords print the value
    for($cv01=0;$cv01<$numberOfKeyWords;$cv01++)
    {
        print $tempKeyWordVariables[$cv01]."\t";
        printf $tmpOut "$tempKeyWordVariables[$cv01]\t"
    }
    print "\n";
    printf $tmpOut "\n"
}

main();

0;
```

# Επεξεργασία αρχείου με κείμενο

- **use ModuleName**
  - Δηλώνονται τα Modules που θα χρησιμοποιηθούν
- **my(\$varName1, \$varName2)**
  - Με τη χρήση του my δηλώνονται μια λίστα από μεταβλητές.
- **ARGV**
  - Λίστα που περιέχει όλες τις παραμέτρους που δόθηκαν από το command line.
  - ARGV[0] ... ARGV[n]. Στην πρώτη θέση δεν βρίσκεται το όνομα του εκτελέσιμου σε αντίθεση με τη C.

# Επεξεργασία αρχείου με κείμενο

- `@listName`
  - Επιστρέφει το μήκος της λίστας
- `sub SubName (parameters) { Code }`
  - Οι συναρτήσεις της Perl είναι απλά block κώδικα
  - Η παράμετροι δεν έχουν όνομα. Απλά δηλώνεται ο αριθμός των scalars που θα περαστούν ως παραμέτροι.
    - Π.χ. `sub subName($$)` παίρνει 2 scalars ως παραμέτρους

# Επεξεργασία αρχείου με κείμενο

- `@_`
  - Περιέχει τις τιμές όλων των scalars που περάστηκαν ως παράμετροι στο subroutine
- `FileHandle`
  - Front-end για τις `IO::*` κλάσεις
  - `$fh = new FileHandle "file", O_WRONLY|O_APPEND;`
    - Επιστρέφει ένα reference στο αρχείο
  - `$fh = new FileHandle "> FOO";`
    - Επιστρέφει ένα reference στο αρχείο FOO όπου μπορούν να γραφτούν δεδομένα.



# Επεξεργασία αρχείου με κείμενο

- **defined EXPR**
  - Επιστρέφει μια boolean τιμή που δείχνει αν το EXPR έχει κάποια τιμή ή είναι undef, "", 0 ή "0".
- **String concatenation**
  - Γίνεται με χρήση της "."
- **Διάβασμα από file**
  - `$line = < fileHandle >`
  - Αν το file handle είναι κενό τότε διαβάζει από τα command line arguments.

# Επεξεργασία αρχείου με κείμενο

- `foreach (@array_list) { body }`
  - Επαναλαμβάνει τον κώδικα στο `body` του loop για κάθε στοιχείο του `array`
  - Το στοιχείο κάθε επανάληψης βρίσκεται στο scalar `$_`
- Γράψιμο σε αρχείο
  - `print FileHandle "text"`

# Προχωρημένα Θέματα

- Είδαμε παραδείγματα επεξεργασίας αρχείων και συμβολοσειρών
- Η Perl παρέχει και πιο προχωρημένες τεχνικές προγραμματισμού
- Αυτά παρέχονται είτε από την γλώσσα είτε από `modules` που αναπτυχθήκαν γύρω από τον πυρήνα του διερμηνέα

# Προχωρημένα Θέματα (συνέχεια)

- Αντικειμενοστραφής Προγραμματισμός
- Multithreading. Υπάρχουν βιβλιοθήκες που υποστηρίζουν την ανάπτυξη multithreading προγραμμάτων
- Βάσεις Δεδομένων
- Τεχνολογίες Διαδικτύου. Ενσωμάτωση κώδικα σε κώδικα HTML

# Πλατφόρμες

- Η Perl αναπτύχθηκε για λειτουργικά συστήματα τύπου UNIX.
- Υπάρχουν και εκδόσεις για άλλα λειτουργικά όπως Windows και MacOS

# Συμπεράσματα

- Η Perl είναι γλώσσα με πολλές δυνατότητες
- Περιλαμβάνει γνωρίσματα για επεξεργασία συμβολοσειρών και άλλων δομών δεδομένων.
- Δεν έχει τους περιορισμούς άλλων scripting languages (tsh, csh, bash)
- Portability λόγω του ότι είναι διερμηνευόμενη
- Διαδόθηκε λόγω των πολλών λειτουργιών που παρέχει στους προγραμματιστές

# Τέλος