«Τεχνογλωσσία VIII»
Εξαγωγή πληροφοριών από κείμενα

Σεμινάριο 4: Εισαγωγή στην γλώσσα Tcl

Ευάγγελος Καρκαλέτσης, Γεώργιος Πετάσης

Εργαστήριο Τεχνολογίας Γνώσεων & Λογισμικού,

Ινστιτούτο Πληροφορικής & Τηλεπικοινωνιών, Ε.Κ.Ε.Φ.Ε. "Δημόκριτος"

Τηλ.: 210-6503197, Fax: 210-6532175, {vangelis, petasis}@iit.demokritos.gr

Ακαδημαϊκό Έτος: 2013 – 2014

Η γλώσσα Tcl/Tk

- Η Tcl βασίζεται σε «συμβολοσειρές»
  - Παλαιότερα ίσχυε ότι «τα πάντα είναι strings»
  - Από το 2000 και μετά, τα “scripts” μεταφράζονται σε “bytecodes”, τα οποία εκτελούνται από τον διερμηνέα της Tcl (Tcl interpreter)
- Δημιουργήθηκε από τον **John Osterhout**, το 1988
  - Στόχος: μια γλώσσα με εύκολη ενσωμάτωση σε εφαρμογές
  - Πλέον: πολλά περισσότερα!
    - rapid prototyping, scripted applications, GUIs and testing

Tcl: Tool Command Language

- Κώδικας: **εντολές**
- Διαδικαστική γλώσσα (procedural language)
  - Έχει λειτουργικά χαρακτηριστικά (functional features)
- Υποστηρίζει αντικειμενοστραφή προγραμματισμό
  - Από την έκδοση 8.6, το πακέτο TclOO περιλαμβάνεται στον πυρήνα της γλώσσας
- Δεν είναι πολύ δημοφιλής σήμερα
- Δύο τρόποι χρήσης:
  - Διαδραστικός διερμηνέας (interactive interpreter)
  - Tcl “scripts”

Διαδραστικός διερμηνέας

```
$ tclsh
% puts $tcl_version
8.6
% puts $tcl_interactive
1
```

Tcl scripts

```
#!/usr/bin/tclsh
# first.tcl
puts "This is a Tcl tutorial“
```

- Εκτέλεση σε λειτουργικό τύπου Unix:

```
$ chmod +x first.tcl
$ ./first.tcl
This is a Tcl tutorial
```

Γραμματική

- Εξαιρετικά απλή (11 κανόνες ακριβώς!)
- Το βασικό στοιχείο είναι οι «λέξεις»
  - Οτιδήποτε δεν περιέχει κενό, οτιδήποτε περικλείεται από "" ή {}
- Οι λέξεις είναι είτε εντολές, ή ορίσματα εντολών
- Η αντικατάσταση παίζει πρωτεύοντα ρόλο

Εντολές (1)

- Οι «προτάσεις» χωρίζονται με χαρακτήρες αλλαγής γραμμής, ή semicolon ";"
- Κάθε «πρόταση» περιέχει
  - Την «εντολή» (command)
  - Η οποία ακολουθείται από λέξεις, που είναι τα ορίσματα (προαιρετικά)
    - Χωρίζονται με κενά
    - Εξαρτώνται από την εντολή
    - `command arg1 arg2 arg3 ...`
    - Κάθε εντολή ερμηνεύεται μέσα στο δικός της περιβάλλον (context)

Εντολές (2)

```
#!/usr/bin/tclsh
puts www.ellogon.org; puts ellogon.com
puts single_word_with_no_spaces
puts Hello
puts "Hello World!"
```

Αντικατάσταση (1)

- Τρεις τύποι αντικατάστασης:
  - Εντολής: `[]`
  - Μεταβλητής: `$`
  - “Backslash”: `\<symbols>`
- Εντολή

```
% puts [expr 1+2]
3
```

- Μεταβλητή

```
set name Jane
puts name
puts $name
```

Αντικατάσταση (2)

- Backslash

```
puts "This was a \"great\" experience“
puts "The \\ character is the backslash character“
puts "20000\b\b miles"
```

- Έξοδος

```
This was a "great" experience
The \ character is the backslash character
200 miles
```

Διάφοροι κανόνες (1)

- Σχόλια: #
  - οτιδήποτε από τον χαρακτήρα # μέχρι την αλλαγή γραμμής, αγνοείται
- Κενοί χαρακτήρες
  - Χωρίζουν λέξεις, συνεχόμενοι υπολογίζονται σαν ένας
- Μεταβλητές: περιέχουν τιμές
  - Ορισμός μεταβλητής: `set όνομα τιμή`
  - Ανάκτηση τιμής: `$όνομα`, `[set όνομα]`
  - Πίνακες: `όνομα(κλειδί)`
    - `$όνομα(κλειδί)`, `[set όνομα(κλειδί)]`

Διάφοροι κανόνες (2)

- Braces: {}
  - Ειδική σημασία: δημιουργούν συμβολοσειρές χωρίς αντικαταστάσεις
    - `puts "Her name is $name"`
    - `puts {Her name is $name}`
  - Δημιουργούν λίστες
    - `set numbers {1 2 3 4 5 6 7}`
- Τελεστής επέκτασης: `{*}λίστα` (από την 8.6 και μετά)
  - Μετατρέπει την λίστα σε διακριτές λέξεις
- Τελειώσαμε!

Εντολές (1)

String Handling
append - Append to variable
binary - Insert and extract fields from binary strings
encoding - Manipulate encodings
format - Format a string in the style of sprintf
prefix - facilities for prefix matching
regexp - Match a regular expression against a string
regsub - Perform substitutions based on regular expression pattern matching
scan - Parse string using conversion specifiers in the style of sscanf
string - Manipulate strings
subst - Perform backslash, command, and variable substitutions

Εντολές (2)

List Handling
concat - Join lists together
join - Create a string by joining together list elements
lappend - Append list elements onto a variable
lassign - Assign list elements to variables
lindex - Retrieve an element from a list
linsert - Insert elements into a list
llength - Count the number of elements in a list
list - Create a list
lrange - Return one or more adjacent elements from a list
lrepeat - Build a list by repeating elements
lreplace - Replace elements in a list with new elements
lreverse - Reverse the order of a list
lsearch - See if a list contains a particular element
lset - Change an element in a list
lsort - Sort the elements of a list
split - Split a string into a proper Tcl list

Εντολές (3)

Control Constructs	
after - Execute a command after a time delay	return - Return from a procedure, or set return code of a script
break - Abort looping command	switch - Evaluate one of several scripts, depending on a given value
catch - Evaluate script and trap exceptional returns	tailcall - Replace the current procedure with another command
continue - Skip to the next iteration of a loop	throw - Generate a machine-readable error
coroutine - Create and produce values from coroutines	try - Trap and process errors and exceptions
error - Generate an error	update - Process pending events and idle callbacks
eval - Evaluate a Tcl script	uplevel - Execute a script in a different stack frame
for - 'For' loop	vwait - Process events until a variable is written
foreach - Iterate over all elements in one or more lists	while - Execute script repeatedly as long as a condition is met
if - Execute scripts conditionally	

Εντολές (4)

Input and Output	
chan - Read, write and manipulate channels I	open - Open a file-based or command pipeline channel
close - Close an open channel	puts - Write to a channel
eof - Check for end of file condition on channel	read - Read from a channel
fblocked - Test whether the last input operation exhausted all available input	refchan - command handler API of reflected channels
fconfigure - Set and get options on a channel	seek - Change the access position for an open channel
fcopy - Copy data from one channel to another	socket - Open a TCP network connection
file - Manipulate file names and attributes	tell - Return current access position for an open channe
fileevent - Execute a script when a channel becomes readable or writable	transchan - command handler API of channel transforms
flush - Flush buffered output for a channel	zlib - compression and decompression operations
gets - Read a line from a channel	

Εντολές (5)

Packages and Source files	Math
load - Load machine code and initialize new commands	expr - Evaluate an expression
loadTk - Load Tk into a safe interpreter	**System Related**
package - Facilities for package loading and version control	cd - Change working
pkg::create - Construct an appropriate 'package ifneeded' command for a given package specification	clock - Obtain and manipulate dates and times
pkg_mkIndex - Build an index for automatic loading of packages	exec - Invoke subprocesses
source - Evaluate a file or resource as a Tcl script	directory exit - End the application
tm - Facilities for locating and loading of Tcl Modules	glob - Return names of files that match patterns
unload - Unload machine code	pid - Retrieve process identifiers
Dictionary Handling	pwd - Return the absolute path of the current working directory
dict - Manipulate dictionaries	time - Time the execution of a script