

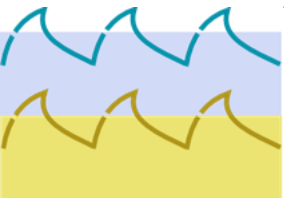
DESERT Underwater: tcl basics

UNWis - Padova (Italy)

29th of January – 3rd of February 2024

**Filippo Campagnaro, Roberto Francescon,
Angela Soldà, Antonio Montanari, Michele Zorzi**

filippo.campagnaro@unipd.it



(O)TCL basics

OTCL <source: wikipedia>

- object oriented extension of Tcl
- Tcl (pronounced "tickle") is a high-level, general-purpose, interpreted, dynamic programming language. Tcl casts everything into the mold of a command, even programming constructs like variable assignment and procedure definition.
- It is commonly used embedded into C applications, for rapid prototyping, scripted applications, and testing.
- DESERT Tcl scripts create and bind the modules and structures written in C++

Variables

set is used to set a variable
set N 5; # this is a comment

\$ is used to access the content
puts prints to standard output

puts "N is equal to \$N"

Associative arrays

opt is an associative array

set opt(string1,string2) 5

```
set opt(nn)                2.0 ;# This is a comment
set opt(starttime)         1
set opt(stoptime)          100000
set opt(freq)              25000.0
set opt(bw)                5000.0
set opt(ack_mode)          "setNoAckMode"
```

Mathematical operations

expr is used to indicate mathematical expressions

```
set opt(txduration) [expr $opt(stoptime) - $opt(starttime)]  
set tot_throughput 0.0  
set tot_throughput [expr $tot_throughput + $throughput]  
set tot_throughput [expr $tot_throughput / $N]  
set pdr [expr $n_rx / $n_tx]  
set x [expr $D * cos($angle) / (2 - cos(2 * $angle))]
```

Note: in case of division, be sure variables are floating numbers

For statement

```
for {set cnt 0} {$cnt < $opt(nn)} {incr cnt} {  
    set tot_nrx [expr $tot_nrx + $n_rx($cnt)]  
    set tot_ntx [expr $tot_ntx + $n_tx($cnt)]  
}  
  
set data [split $file_data "\n"]  
foreach line $data {  
    puts $data  
}
```

Parse text files with regepr

```
set opt(waypoint_file) "../dbs/wp_path/rov_path.csv"
set fp [open $opt(waypoint_file) r]
set file_data [read $fp]
set data [split $file_data "\n"]
foreach line $data {
    if {[regexp {^(.*),(.*),(.*),(.*)$} $line -> t x y z]} {
        puts "time = $t"
        puts "position = $x, $y, $z"
    }
}
```

rov_path.csv

```
200.000000,100.000000,0.000000,-15.000000  
230.869038,99.950656,24.868989,-15.000000  
260.175886,99.802673,48.175367,-15.000000  
286.456727,99.556196,68.454711,-15.000000  
308.438527,99.211470,84.432793,-15.000000  
325.120561,98.768834,95.105652,-15.000000
```

The format of each row is:

<time>,<x>,<y>,<z>

Functions (procedures)

A subset of global variables can be captured by the function

```
proc connectNodes {id1 id2} {  
    global ipif ipr portnum  
    #body of the function here  
}  
connectNodes 5 6
```