TCL Scripting for OPENLANE

Ahsen Tahir
Electrical Department
University of Engineering and Technology, Lahore

TCL Script File Execution

Make a file with the vim editor and tcl extension

```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ vi filename.tcl
```

To invoke the tclsh interpreter when executing file write #!/usr/bin/tclsh (where # is for comment) as first line

```
1 #!/usr/bin/tclsh
2
3 puts "hello"
```

Execute on command prompt

```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ filename.tcl
filename.tcl: command not found
```

sorry the filename.tcl is not in the path such as /usr/bin etc.
Please use ./filename.tcl (./ means present directory just like ../ means parent directory)

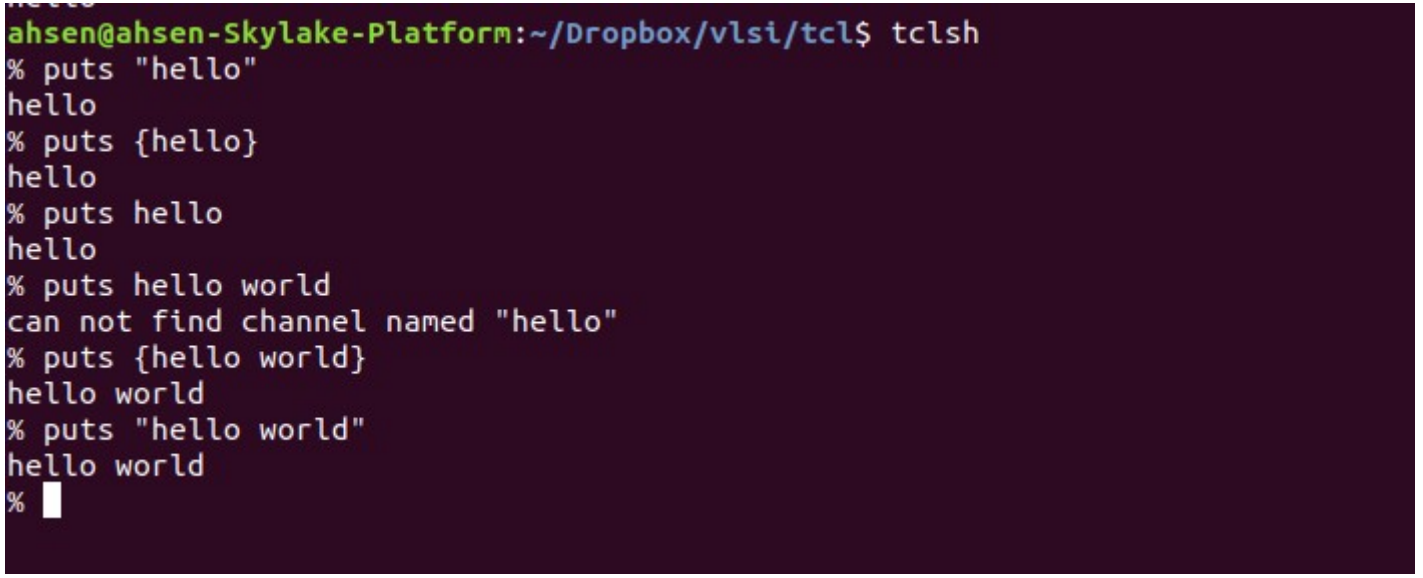
```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ ./filename.tcl
bash: ./filename.tcl: Permission denied
```

Linux needs execution permissions for the file through **chmod** command

```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ chmod 777 filename.tcl
```

```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ ./filename.tcl
hello
```

TCL Execution on Shell Interpreter



Assigning Variables

- Assign a variable with set command
  - set a 1
  - set b flower
  - set c 2.2
- Reference variable with $, e.g. $a, $b, $c, puts $a (/r for carriage return, /t for tab), puts “$a” (not {$a})

```
ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ tclsh
% set a 1
1
% puts $a
1
% set b 2
2
% puts $a*$b
1*2
%
```

Mathematical Expressions

- expr {$a+$b}
- expr {$a == 1} or expr {$a eq 1}
- Also eq, ne, !, <=, >=, << , >>
- Arrays
  - set a(1) 10
  - set a(2) 7
  - set a(3) 17
  - set b 2
  - puts "Sum: [expr {$a(1)+$a($b)}]"

Loops

- for {set i 0} {$i < 10} {incr i} {

  puts $i

  }
- while {$i < 10} {

  puts $i

  set i [expr {$i + 2}]

  }

If - else

- set x 3
- if {$x == 2} {
  puts "x is 2"
  } elseif {($x > 2) && ($x <= 4)} {
  puts "x is greater than 2 and less than equal to 4"
  } else {
  puts "x is greater than 4"
  }

Procedure/function in TCL

- proc procedureName {arguments} {
  body
  }
- proc sum {arg1 arg2} {
    set x [expr {$arg1 + $arg2}] (or return [expr {$arg1 + $arg2}])
    return $x
  }

- puts "[sum 2 3]"
- set value [sum 2 3]

Associative Arrays

- Array
  - set a(1) hello
  - set a(2) world
- Associative Array
  - set a(age) 12
  - set a(car) toyota
- parray a
  a(age) = 12
  a(car) = toyota
- array names a
  age car
- info exists a(age)
  1

```
(base) ahsen@ahsen-Skylake-Platform:~/Dropbox/vlsi/tcl$ tclsh
% set a(age) 12
12
% set a(car) toyota
toyota
% parray a
a(age) = 12
a(car) = toyota
% array names a
age car
% info exists a(age)
1
% info exists a
1
% info exists a(address)
0
%
```

TCL Array: env

- Array env is used for environment variables by TCL
  - e.g. env(PATH), env(USER), env(HOSTNAME) and openlane flow variables
  - env has a global name space and can also be written ::env()
- Try **parray env** to get the list and **puts [array names env]** to get index
- Used in openlane config files for variables

  e.g. set ::env(DESIGN_NAME) alu, set ::env(CLOCK_PERIOD) "10.000"

  set ::env(CLOCK_PORT) "clk"

- Check if a particular env variable (index→ DISPLAY) exists
  if { [info exists ::env(DISPLAY)] }
- Other global variables argv → Tcl list of arguments to tclsh or wish.

Misc. Built-in Commands

- puts [info script] → ./filename.tcl
- puts [file normalize [info script]] → /*home/*ahsen/vlsi/filename.tcl
- puts [file dirname [file normalize [info script]]]→ /*home/*ahsen/vlsi
- lappend will append new path to internal paths maintained by TCL

  e.g. lappend ::auto_path $::env(OPENLANE_ROOT)

  e.g. lappend ::auto_path “$::env(OPENLANE_ROOT)/scripts”
- Execute commands/programs in tcl

  e.g. exec tclsh >&@stdout

  (>&@stdout → sends standard output from the last command and standard error from all commands to standard output)
- glob command is used to match file names and outputs a list

  glob *.v → design_2.v design_1.v