



DELCPG012

Arquitetura ASIC e FPGA

Aula 07: TCL Scripting



UFSM



UnB



Universidade Federal
de Campina Grande



UNIVERSIDADE
FEDERAL DO CEARÁ



Objetivos

- Introdução ao TCL
- Comandos básicos
- Exemplos de aplicação em VLSI

Introdução ao TCL

Duas partes do aprendizado do Tcl:

1. Regras de sintaxe e substituição:

- As substituições são simples, mas podem ser confusas no início.

2. Comandos incorporados:

- Pode ser aprendido individualmente, conforme necessário.
- As estruturas de controle são comandos, não a sintaxe da linguagem.

Introdução ao TCL

Tcl script =

- Sequência de comandos.
- Comandos separados por novas linhas, ponto e vírgula.

Tcl command =

- Uma ou mais palavras separadas por espaço em branco.
- A primeira palavra é o nome do comando, as outras são argumentos.
- Retorna o resultado da string.

Exemplos:

```
set a 22; set b 33
```

```
set a 22
```

```
set b 33
```

Introdução ao TCL

O Tcl Parser:

- Divide os comandos em palavras.
- Faz substituições.
- Não interpreta os valores das palavras.

O Command Procedure:

- Interpreta palavras.
- Produz o resultado da string.

As substituições realizadas pelo analisador são de “uma etapa”; os valores substituídos não são verificados novamente em busca de texto que se pareça com substituições. Alguns comandos específicos ([eval], [expr], [subst], ...) executam outra rodada de substituição em seus argumentos.

Introdução ao TCL

Argumentos

- O Parser não atribui nenhum significado aos argumentos (citando por padrão, a avaliação é especial):

C:

x = 4; y = x + 10;
Resultado de y é 14

Tcl:

set x 4; set y x+10
Resultado de y é x+10

- Diferentes comandos:

set a 122
expr 24/3.2
eval "set a 122"
string length CI-Inovador

Introdução ao TCL

Substituição de variáveis

- Sintaxe: \$varName
- O nome da variável é composto por letras, dígitos, sublinhados.
- Pode ocorrer em qualquer lugar em uma palavra.

Sample command Result

set b 66	66
set a b	b
set a \$b	66
set a \$b+\$b+\$b	66+66+66
set a \$b.3	66.3
set a \$b4	no such variable

Introdução ao TCL

Substituição de comandos

- Sintaxe: [script]
- Avalia o script e substitui o resultado.
- Pode ocorrer em qualquer lugar dentro de uma palavra.

set b 8

#-> 8

set a [expr {\$b+2}]

#-> 10

set a "b-3 is [expr {\$b-3}]"

#-> b-3 is 5

Introdução ao TCL

Controle da estrutura das palavras

- As palavras são interrompidas por espaço em branco e ponto e vírgula exceto:
 - As aspas duplas evitam quebras:

```
set a "x is $x; y is $y"
```

- As chaves evitam quebras e substituições:

```
set a {[expr {$b*$c}]}
```

- As barras invertidas citam caracteres especiais:

```
set a word\ with\ \$\ and\ space
```

- As substituições não alteram a estrutura das palavras:

```
set a {two words}  
set b $a
```

Introdução ao TCL

Expressões

- C-like (int e double), suporte extra para operações com strings.
- Comando, a substituição de variáveis ocorre dentro de expressões.
- Usado em **expr**, outros comandos.

Sample command	Result
set a 10	10
set b 5	5
expr (\$b*4) - 3	17
expr \$b <= 2	0
expr \$a * cos(2*\$b)	-8.3907
set a Bill	Bill
expr {\$a < "Anne"}	0

Introdução ao TCL

Listas

- Zero ou mais elementos separados por espaço em branco:

red green blue

- Chaves e barras invertidas para agrupamento:

a b {c d e} f

one\ word two three

- Alguns comandos para listas:

concat	lindex	llength	lsearch	foreach	linsert
lrange	lsort	lappend	list	lreplace	

- Exemplo

lindex {a b {c d e} f} 2

#-> c d e

lsort {red green blue}

#-> blue green red

Introdução ao TCL

- Utilização

- **Creating a List**

`set` listName { item1 item2 item3 .. itemn }

`set` listName [list item1 item2 item3]

`set` listName [split "items separated by a character" split_character]

- **Appending**

`append` listName split_character value

or

`lappend` listName value

- **Sorting**

`lsort` listname

- **Length of List**

`llength` listName

Introdução ao TCL

- Utilização

- **List Item at Index**

`lindex` listName index

- **Insert Item at Index**

`linsert` listName index value1 value2..valuen

- **Replace Items at Indices**

`lreplace` listName firstindex lastindex value1 value2..valuen

- **Set Item at Index**

`lset` listname index value

- **Transform List to Variables**

`lassign` listname variable1 variable2.. variablen

Introdução ao TCL

Estruturas de controle

- Aparência semelhante a C.
- Apenas comandos que usam scripts Tcl como argumentos.

Comandos

if for switch break foreach while eval continue

Exemplo

```
puts "Hello world!"  
set a {blue red green violet orange}  
set b {}  
set i [expr [llength $a] - 1]  
puts "value of i = $i"  
while {$i >= 0} {  
    lappend b [lindex $a $i]  
    incr i -1  
}  
puts $b
```

Introdução ao TCL

- Utilização

→ if

```
if {boolean_expression} {  
    # statement(s) will execute if the Boolean expression is true  
}
```

```
1  #!/usr/bin/tclsh  
2  set a 10  
3  #check the boolean condition using if statement  
4  if { $a < 20 } {  
5  # if condition is true then print the following  
6  puts "a is less than 20"  
7  }  
8  puts "value of a is : $a"
```

Introdução ao TCL

- Utilização

→ **if ...else if ... else**

```
if {boolean_expression 1} {  
    # Executes when the boolean expression 1 is true  
}  
elseif {boolean_expression 2} {  
    # Executes when the boolean expression 2 is true  
}  
elseif {boolean_expression 3} {  
    # Executes when the boolean expression 3 is true  
}  
else {  
    # executes when the none of the condition is true  
}
```

```
1  #!/usr/bin/tclsh  
2  
3  set a 100  
4  
5  #check the boolean condition  
6  if { $a == 10 } {  
7      # if condition is true then print the following  
8      puts "Value of a is 10"  
9  } elseif { $a == 20 } {  
10     # if else if condition is true  
11     puts "Value of a is 20"  
12 } elseif { $a == 30 } {  
13     # if else if condition is true  
14     puts "Value of a is 30"  
15 } else {  
16     # if none of the conditions is true  
17     puts "None of the values is matching"  
18 }  
19  
20 puts "Exact value of a is: $a"
```


Introdução ao TCL

- Utilização

→ Switch

```
switch switchingString {  
  matchString1 {  
    body1  
  }  
  matchString2 {  
    body2  
  }  
  ...  
  matchStringn {  
    bodyn  
  }  
}
```

```
1  #!/usr/bin/tclsh  
2  
3  set grade B;  
4  
5  switch $grade {  
6    A {  
7      puts "Well done!"  
8    }  
9    B {  
10     puts "Excellent!"  
11   }  
12  
13   C {  
14     puts "You passed!"  
15   }  
16   F {  
17     puts "Better try again"  
18   }  
19   default {  
20     puts "Invalid grade"  
21   }  
22 }  
23 puts "Your grade is $grade"
```

Introdução ao TCL

→ While

```
while {condition} {  
    statement(s)  
}
```

```
1  #!/usr/bin/tclsh  
2  
3  set a 10  
4  
5  #while loop execution  
6  while { $a < 20 } {  
7      puts "value of a: $a"  
8      incr a  
9  }
```

Introdução ao TCL

→ For

```
for {initialization} {condition} {increment} {  
    statement(s);  
}
```

```
1  #!/usr/bin/tclsh  
2  
3  # for loop execution  
4  for { set a 10} {$a < 20} {incr a} {  
5      puts "value of a: $a"  
6  }  
7
```

Introdução ao TCL

→ Foreach

```
foreach varlist list {  
    statement(s);  
}
```

```
1  #!/usr/bin/tclsh  
2  
3  set ListName [list orange red blue violet green]  
4  
5  # foreach loop execution  
6  foreach a $ListName {  
7      puts "value of a: $a"  
8  }
```

Introdução ao TCL

Procedures

- `[proc]` Define um procedimento:

```
proc sub1 x {expr $x-1}
```

```
      ^      ^      ^  
      |      |      |  
      name   |      body  
              |  
list of argument names
```

Os procedimentos se comportam exatamente como os comandos built-in:

```
sub 3 → 2
```

Os argumentos podem ter padrões

```
proc decr {x {y 1}} {  
    expr $x-$y  
}
```

Introdução ao TCL

→ Listas de argumentos de comprimento variável:

```
proc sum args {  
    set s 0  
    foreach i $args {  
        incr s $i  
    }  
    return $s  
}
```

```
sum 1 2 3 4 5
```

```
#-> 15
```

```
sum
```

```
#-> 0
```

Introdução ao TCL

→ **proc:**

```
1  #!/usr/bin/tclsh
2
3  proc helloWorld {} {
4      puts "Hello, World!"
5  }
6
7  proc add {a b} {
8      return [expr $a+$b]
9  }
10
11 proc avg {numbers} {
12     set sum 0
13     foreach number $numbers {
14         set sum [expr $sum + $number]
15     }
16     set average [expr $sum/[llength $numbers]]
17     return $average
18 }
19
20 helloWorld
21 set s [add 10 30]
22 puts $s
23 puts [avg {70 80 50 60}]
```

Introdução ao TCL

- **Abrindo e fechando arquivos**

- open fileName accessMode (r, w, a)
 - `set fp [open "fileName" w]`
- close fileName
 - `close $fp`

- **Escrevendo no arquivo**

- `puts $filename "text to write"`

- **Lendo o arquivo**

- `set file_data [read $fp]`

```
1  #!/usr/bin/tclsh
2
3  # Writing file
4  set fp [open "input.txt" w]
5  puts $fp "CI Inovador"
6  close $fp
7
8  # reading file
9  set fp [open "input.txt" r]
10 set file_data [read $fp]
11 puts $file_data
12 close $fp
```