Using NS2 Presentation 2

tcl
and
otcl

tcl and otcl

- tcl is a language designed for use by an application developer
  - Can be “embedded” (made “part of”) an application
  - Could be used by an application in various ways, e. g. to allow a user to supply a custom initialization for the application
- otcl is tcl with “object-oriented” extentions
- ns2 uses otcl for the simulation programmer to create the network objects in memory and to insert initial events into the event queue

Scripting language

- tcl is a scripting language; it is executed by an interpretter
- Syntax is similar to simple scripting languages like a Unix/Linux shell, e. g. bash
- A statement is a series of “words” (non white space characters) separated by white space (spaces and tabs); “words” may be composed of most other characters
- The first word on a line must be a command name; the rest of the words are arguments

Variables

- The interpreter keeps a table of (variable) name/value pairs
- The set command causes the interpreter to set or modify the value of a name in the table
- Syntax:
  - `set name value`
  - Where name and value are single words

Variable examples

- Example:
  - `set a 1`
  - `set b 2`
  - `set c a+b`
  - `puts “$a $b $c”`
- Produces:
  - `1 2 a+b`
- Remember: they're just words!

Those pesky dollar signs

- A tcl program has many “dollar signs” (“$”s)
  - I wish they were money
- A “$” followed by a name causes the interpreter to just replace them by the value from the name/value table
- Interpreter algorithm
  - Do the replacements in the line FIRST
  - THEN treat the MODIFIED line as a command

Variable substitution example

- Example:
  - set v1 1
  - set v2 2
  - set v3 v1
  - set $v3 $v2
  - puts “$v1 $v2 $v3”
- Produces:
  - 2 2 v1

Command substitution

- Uses brackets: “[“ and “]”
- Interpreter algorithm:
  - FIRST do the command inside the brackets
  - Replace the entire bracketed part with the value returned by the command
  - Do the outer command
  - In the case of multiple nested commands, work from the inner to the outer

Command substitution example

- Example:
  - `set r1 [ set r2 [ set r3 25 ] ] ]`
  - `set r4 [ expr $r1 + 1 ]`
  - `puts “$r1 $r2 $r3 $r4”`
- Notes:
  - set command returns value assigned
  - expr command evaluates an expression and returns the result
- Produces:
  - `25 25 25 26`

What about otcl

- Introduces classes
  - Programmer can introduce their own classes, but we will mostly use those provided by ns2
  - As usual, classes specify instance variables and methods
- Can “instantiate” a class, which gives an object
  - Each object contains its own set of values for the instance variables
  - Objects are used to represent the network, e. g. nodes, links, etc.

Object creatiion

- The “new” statement makes an object
  - `new` *`ClassName`*
  - The value returned is a “reference” to the created object
  - The reference may be needed in a subsequent statement, e. g. creation of a link requires specification of the two nodes to which it connects
  - “`Simulator`” is a class, and we need to make an instance of this class at the beginning of an ns2 program
  - Many ns2 methods return a class reference

Method invocation

- To invoke a method defined for the class of an object use:
  - *`ObjectReference methodName arguments`*
  - Note that object references are generated automatically by otcl when an object is created; because we don't know what the object reference will be in advance, we usually need to save the object reference value in a variable and use variable substitution

otcl example

- Example:
  - `set ns [new Simulator]`
  - `puts "$ns"`
  - `set n0 [$ns node]`
  - `set n1 [$ns node]`
  - `set link0 [$ns duplex-link $n0 $n1 10Mb 1ms DropTail]`
  - `set n2 [$ns node]`
  - `puts "$n0 $n1 $n2 $link0"`

otcl example result

- Produces:
  - _o3
  - _o10 _o13 _o28
- Note: the duplex-link method of the simulator returns a null value, not an object reference to a link object!

Ns2 available classes and methods

- There are MANY methods provided by the classes provided by ns2
  - Look at examples
  - Look in the reference manual
  - Search google