TLC-TK on HPC

What is TLC-TK?

TCL (Tool Command Language) is a high-level, dynamic programming language that is designed to be easy to learn and use. It is often used for scripting, embedded applications, and GUI development. TCL is an interpreted language, which means that the code is executed line-by-line as it is read, rather than being compiled into machine code. This makes it easy to develop and debug code quickly.

TK (Tool Kit) is a graphical user interface (GUI) toolkit that provides widgets and tools for building graphical user interfaces. TK is built on top of TCL and provides a set of tools and widgets that make it easy to create complex, interactive graphical interfaces. TK widgets include buttons, menus, text boxes, and scrollbars.

Together, TCL and TK are often referred to as "TCL/TK". TCL/TK is a popular choice for developing GUI applications, particularly in the scientific and engineering communities. TCL/TK has been used in a wide range of applications, including electronic design automation, scientific visualization, and simulation. TCL/TK is also often used in embedded systems, as its small footprint and low overhead make it a good choice for systems with limited resources.

Links:

Official Website

Manual

Versions Available:

The following versions are available on the cluster:

- `TLC-TK 8.6.3`

How to load TLC-TK?

To load TLC-TK, use the following commands:

```
#Load the TLC-TK module
module load tcltk/8.6.3
```

To verify if the module is loaded correctly, use the following command,

```
# List all the module loaded in the environment
module list
```

How to use TLC-TK?

To use Tcl/Tk, one first needs to install the Tcl/Tk interpreter on their system. Then, a Tcl script can be written using a text editor and saved with a .tcl file extension. The script can then be executed from the command line using the tclsh or wish command followed by the name of the script file. Within the script, the Tk toolkit can be used to create graphical user interfaces using various widgets such as buttons, labels, and text boxes. The script can also interact with the system through various built-in commands and external packages.

Here is a sample script that makes a simple analog and digital clock,

```
#!/usr/bin/env tclsh
package require Tk

proc every {ms body} {eval $body; after $ms [info level 0]}

proc drawhands w {
    $w delete hands
```

```
    set secSinceMidnight [expr {[clock sec]-[clock scan 00:00:00]}]
    foreach divisor {60 3600 43200} length {45 40 30} width {1 3 7} {
       set angle [expr {$secSinceMidnight * 6.283185 / $divisor}]
       set x [expr {50 + $length * sin($angle)}]
       set y [expr {50 - $length * cos($angle)}]
       $w create line 50 50 $x $y -width $width -tags hands
    }
}
proc toggle {w1 w2} {
    if [winfo ismapped $w2] {
        foreach {w2 w1} [list $w1 $w2] break ;# swap
    }
    pack forget $w1
    pack $w2
}
#-- Creating the analog clock:
canvas .analog -width 100 -height 100 -bg white
every 1000 {drawhands .analog}
pack .analog

#-- Creating the digital clock:
label .digital -textvar ::time -font {Courier 24}
every 1000 {set ::time [clock format [clock sec] -format %H:%M:%S]}

bind . <1> {toggle .analog .digital}
```

To execute a script, add or execution rights to file or use tclsh command,

```
#execute a script
tclsh example.tcl
```

Where to find help?

If you are confused or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp