Tcl Bounties

FlightAware

November 16, 2016

Tcl Bounties

- FlightAware is offering a number of bounties for various enhancements, fixes, etc, for Tcl and/or Tcl extensions
- Donal referred to this as a series of Grand Challenges.
- Recognizing a golden era of progress in the Tcl language was when Sun funded work on it, we see an opportunity to provide leadership in causing the evolution of Tcl.
- We are completely serious.
- Bounties we are offering run to tens of thousands of dollars.

Rules for Bounties

Rules for Bounties

- All code must be released under the BSD license.
- For bounties $10,000 and over, we'll pay 50% of the bounty will be paid out when the code is accepted into the Tcl core and the remaining 50% when it appears in a release of Tcl.

Rules for Bounties

- For bounties under $10,000, the bounty will be paid when the code is accepted into the Tcl core, or if not part of the core, accepted by the package maintainer.
- For bounties $10,000 and over, we'll pay 50% of the bounty will be paid out when the code is accepted into the Tcl core and the remaining 50% when it appears in a release of Tcl.

Rules for Bounties

- The first person or team to succeed wins the bounty.
- If you succeed in fulfilling the conditions for receiving a bounty as a team then the team has to apportion the bounty among themselves; we are not getting involved in any disputes over who deserves what.

Rules for Bounties

- We request that people or teams publicly announce their intention to pursue a bounty to reduce the likelihood of wasted work, hard feelings, etc.
  - That being said we understand that some people or teams may announce and not succeed, so the fact that someone has announced they are pursuing a bounty does not prevent others from pursuing it as well.

Rules for Bounties

- If due to the nature of their employment someone is not allowed to accept a bounty or their share of a bounty, they can assign their share to others on their team, to a charity of their choosing, or to the Tcl Community Association.
- FlightAware employees can participate but not double dip, i.e. don't work on it at work.

Bounties - scotty

- $1000 to update Scotty to modern TEA standards, compiling properly under FreeBSD and Mac OS X, with stub support
- $1000 to fix bugs in Scotty's UDP stuff -- "configure/cget" of config vars working and configured host and port to be used if not specified in "send"

SO_REUSEPORT on server sockets

- This would allow multiple programs to open the same port from a server socket.
- It would be a switch to the socket command.
- It looks pretty easy.
- $2500

Make TclX's profiler work properly with Tcl 8.6

- (currently it crashes Tcl)
- this will need Tcl core changes to support it
- $2,500

"array default arrayName value"

- Default values for arrays "array default arrayName value", causes that value to be returned any time an attempt is made to access an element of the array that isn't present.
- $2,500

array foreach

- array foreach varName {code}
- $2,500

A more legit way to get a list of all the source files loaded by a package.

- $2,500

Bounties - tclreadline

- $2,500 to make tclreadline able to recognize iTcl objects and do tab completion for method names, tab completion for variables if indicated by the presence of cget or configure
- tclOO too

Cleanup of tcltls

- support all TLS versions.
- fix hangs in protocol negotiation.
- fix background errors that don't provide an error message.
- test and get it working with LibreSSL -- drop SSLv2, SSLv3, etc.
- plus light maintenance for a year
- $5,000

Speed up clock format and clock scan

- 2X for $5,000
- 4X for $10,000
- 10X for $20,000

A reasonable C API for enumerating an array

- Without using any Tcl code
- $5,000

Make TclX's signal trap handlers safe to use with threaded Tcl

- currently they are prone to deadlock as the signal can be delivered to the select thread and this will self-deadlock if it happens at the wrong time
- this will need Tcl core changes to support it
- $5,000

Stop Tcl from eating child process exit status gratuitously

- currently if you want to retrieve child status for subprocesses yourself asynchronously, there are a bunch of things you must avoid, including "open |foo" even if it's a completely unrelated child
- Tcl core should either not call waitpid for children it is not directly managing, or it should have a mechanism for preserving the return statuses it consumes via waitpid so the app can retrieve them when needed

Stop Tcl from eating child process exit status gratuitously

- this makes any non-trivial communication scheme with subprocesses painful (see piaware's fa_sudo to see the lengths you must go to, and then everything must use that and avoid the core pipe functionality)
- $5,000

Call-by-name syntactic sugar for proc definitions that would obviate most uses of upvar

- something like... proc find_flightplan_from_position {*flightplan *position} {}
- ...that would replace proc find_flightplan_from_position {_flightplan position} ... upvar $flightplan flightplan $_position position, etc
- $10,000

Support for epoll()/kqueue() to replace select() in socket handling

- This could vastly reduce the overhead for Tcl programs that have thousands of sockets open.
- There was a google summer of code project to do this that never got integrated and might not have been completed but could be a starting point.
- $10,000

Revive the Tcl Pro debugger

- $20,000

A first class, high-performance, non-hackish way to do named parameters

- Invoke a proc by naming the variables and their values.
- $20,000

Bounties - Tcl core

- $10,000 for Tcl to run 2X faster than Tcl 8.6 for a benchmark program TBD
- $100,000 for Tcl to run 10X faster than Tcl 8.6
  - I need help with a benchmark.

Bounties - going from here

- https://github.com/flightaware/Tcl-bounties
- We are willing to fund additional work on a case-by-case basis.