

Re: vwait behavior

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2008-03/msg00224.html>

- *From:* Alexandre Ferrieux <alexandre.ferrieux@xxxxxxxx>
 - *Date:* Thu, 6 Mar 2008 14:14:54 -0800 (PST)
-

On Mar 6, 8:52 pm, Donald Arseneau <a...@xxxxxxxx> wrote:

On Mar 5, 8:46 pm, rfwoode...@xxxxxxxx wrote:

I've been wrestling with this problem for several days now. I run the following loop on a linux (red hat) web server and it will run fine for sometimes up to 2 or 3 days before it crashes. Most often it runs for less than 48 hours. (I know it was not due to a server crash – according to uptime.)

Is there something wrong with the way I'm implementing the pause proc that causes some weird behavior?

Thanks for any help!

```
# pauses for n milliseconds
proc pause { {msecs 1000} } {
    global gv;
    after $msecs "set vpause 1"
    vwait vpause
}
```

My guess is you are crashing due to stack overflow. The problem is that Tcl events / event handlers are not independent, but are stacked;

Re: vwait behavior

search for "recursive event loop". If some event occurs while you are paused, the pause cannot return when its time is up, but can only return when the nested event handler is complete. If you have some fraction of events that take a long time to handle or invoke [pause] then the stack will keep growing.

Doesn't apply here since there's no way for vwaits to nest (assuming there is no other code than the one posted): the only place [vwait] is called is within [pause] which is called in a synchronous [while].

Doug: This one beats me... Two questions:

- can you get the standard error of the process ? empty ?
- can you run strace -tt -o somefile yourprocess, until it exits, and post back the end of the trace ?

-Alex

.