Subject: Re: HANDLE_FREE: when to use? is it necessary?
Posted by chase on Mon, 05 Aug 1996 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Stein" == Stein Vidar (UiO) <steinhh@cdsa2.nascom.nasa.gov> writes:
In article <4tqos5$inl@post.gsfc.nasa.gov> steinhh@cdsa2.nascom.nasa.gov (Stein Vidar (UiO)) writes:

Stein> In article <77g26a8wtl.fsf@custer.jhuapl.edu>, chase@custer.jhuapl.edu (Chris Chase SRM) writes:
Stein> |> I use IDL version 4.0.1 on various UNIX systems (HPUX, IRIX, SunOS).
Stein> |> "help,/handle" gives some revealing statistics about handles. It

Stein> Thanks! I didn't know about the help,/handle switch. It works on
Stein> IDL 3.6.1 as well.

Stein> |> Because top-level handles are like a global variable there is really
Stein> |> no reason to have thousands of them available simultaneously - it would
Stein> |> be like a C program with thousands of different variables which would
Stein> |> overwhelm most C compilers.
Stein> |>

Stein> I disagree -- for a handle to act like a global variable, you'd need
Stein> a global variable to store the handle number.

I was comparing only top level handles to global variables, not child or sibling handles. Only top level handle IDs need to be saved in a variable.

Stein> You could have, e.g., a linked list, a linked list of lists, or a
Stein> tree, or whatever, using thousands of handles. But I guess RSI
Stein> people aren't used to that... The C program equivalent would need
Stein> one global pointer.

I was mistaken about the handle table. I assumed that "help,/handle" gave statistics on a table containing only top level handles (which are like globals). However, after running a test, I found that child handles (thousands of which would be legitimate and common) seem to take the same resources in the handle level table. A handle version of a single linked list with 10,000 elements degrades performance as much as having 10,000 top level handles.

I would have thought that child handles would be looked up directly from pointers in parent and sibling handles. Then only top level handle IDs would need to be in a fast lookup table. [The only reason for such a table would be for a user interface to lookup top level handles via a handle ID tag when a direct reference is unavailable.

As I mentioned previously, handles really need to be a distinct data type like structures and not treated as long integers. In this way handles references could be tracked and unreferenced handles garbage collected. In this case, a variable that is a "handle" data type serves the same function as a handle ID tag.]

Chris

--

==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu