Subject: Re: Memory allocation problem:
Posted by David Kastrup on Fri, 27 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

jyli@anchor.gsfc.nasa.gov (Jason Li) writes:

> David Fanning (davidf@dfanning.com) wrote:
>
> : This is a result of IDL being written in C and using the C library
> : functions (malloc and free) for memory allocation. In most C libraries,
> : memory that is freed is NOT returned to the operating system. The C
> : program retains this memory and will reuse it for future calls to malloc
> : (assuming that the new allocation will fit in the freed block).
>
> Retains this memory for how long?
> (1) life of a subroutine
> (2) life of a main routine
> (3) life of an IDL session

(3). However, glibc, the library used in newer versions of Linux (for exmaple) will allocate larger chunks of memory with a different mechanism making it possible to return such larger pieces of memory to the operating system the moment they are freed, regardless of allocation order.

The main advantage of this scheme is not as large as one would naï¿½vely be let to think, as unused memory tends to get swapped out, anyway, when memory gets scarce. But at least this avoids more useful memory pieces to be swapped out instead (causing a bit of thrashing) and it reduces the general impact on swap space related resources.

If you are concerned about proper resource utilization, you should choose your operating system accordingly.

--
David Kastrup Phone: +49-234-700-5570
Email: dak@neuroinformatik.ruhr-uni-bochum.de Fax: +49-234-709-4209
Institut fï¿½r Neuroinformatik, Universitï¿½tsstr. 150, 44780 Bochum, Germany