Subject: Re: Q: Efficient Memory handling and deallocation
Posted by rutledge on Fri, 05 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <D82wnn.K29@hpl.hp.com>, peter@hpl.hp.com (Peter Webb) writes:
>
> Would a C program, using malloc(), perform any differently? That is, if
> I had a program that malloc'ed a big block, then free'd it, then
> malloc'ed a little block, then started over, would I see the same
> behavior as I see from IDL?
>
> If there is no difference, then it really is an OS problem/feature, and
> RSI would have to add their own memory management. If not, then maybe
> RSI could do something about it.

Most definitely in SunOS, free() results in de-allocating memory from the program heap, making it available to other users. I have been unable to get my applications to do this, and it causes major slowing down of programs for me (for instance, I need at one point, LOTS of data, which gets sorted out, and binned, after which I de-allocate it, but then -- I get stuck paging through this later because the 100M I allocated is still in the heap, even though I de-allocated it).

Paul Probert (probert@uwmfe.neep.wisc.edu) wrote:
: But we figured out, as you did, that IDL
: doesn't deallocate the memory. One workaround is, at the beginning of
: your program, create and then immediately delete an array 2 or 3 times
: the size of your needs, and this will leave a hole big enough for many
: future reallocations.

True enough, but my problem is that I want the heap to be freed, so that I don't have to page through at 150M heap (when my computer only has 96M). Any suggestions?

Bob