

Re: The dangers of replacing malloc()

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2003-06/0105.html>

From: D. J. Bernstein (*djb_at_cr.yp.to*)

Date: 06/26/03

Date: 26 Jun 2003 21:05:07 -0000

To: freebsd-performance@freebsd.org

Terry proposed that I obtain the desired sequence of `sbrk()` calls by writing my own version of `malloc()`. I explained how this can destroy data; it isn't portable. Terry now makes the rather strange claim that POSIX guarantees the portability of the resulting programs—even though POSIX doesn't specify `sbrk()` or various other functions that I use.

Terry Lambert writes:

- > *Memeory allocation in the C library is *defined by standards*,*
- > *all of which require the ability to replace memory allocation and*
- > *freeing functions with equivalents, *as a unit*.*

You obviously aren't claiming that POSIX requires `_all_` system functions to use a replacement `malloc()/realloc()/free()` library for allocation. That would prohibit `sbrk()` itself, for example.

Are you trying to say that POSIX requires all `_POSIX functions_` to do their memory allocation via `malloc()/realloc()/free()`? This has no relevance to anything I said. We aren't talking about programs that restrict themselves to the POSIX functions.

- > > *Suppose the OS distributor decides that `valloc()` or `xyzalloc()` should do*
- > > *its own thing, rather than calling `malloc()`.*
- > *Then that OS distributor's OS no longer complies with standards.*

I already showed you code demonstrating that the Linux `valloc()` works this way. If, as you claim, there's a ``standard" prohibiting the Linux behavior, then that ``standard" is not useful for people who care about real-world portability. Anyway, I see no evidence supporting your claim.

- > *If you are in non-compliance with the Intel Application Binary Interface*
- > *specification, you should expect to *minimally* be required to relink,*
- > *recompile, or have to modify your program source code, each time the*
- > *OS major version number changes*

If you link statically, upgrades can break your `_source code_?`

Wow. I never realized that the `-static` option had such power. Will it

freebsd-performance: Re: The dangers of replacing malloc()

also cause hair to grow on your palms?

---D. J. Bernstein, Associate Professor, Department of Mathematics,
Statistics, and Computer Science, University of Illinois at Chicago

freebsd-performance@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@freebsd.org"