

Re: ten thousand small processes

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2003-06/0087.html>

From: Jeff Roberson (jroberson_at_chesapeake.net)

Date: 06/25/03

Date: Tue, 24 Jun 2003 20:45:24 -0400 (EDT)

To: "D. J. Bernstein" <djb@cr.jp.to>

I understand you have some history with the project. As such I believe you deserve a more thorough answer. More replies inline.

On 23 Jun 2003, D. J. Bernstein wrote:

> *alloca() is inadequate. Memory allocation doesn't always follow a stack
> pattern. I certainly can't predict memory use at the top of main().*
>
> *sbrk() is dangerous. Suppose a system call that I'm using turns into a
> library routine that calls malloc()---this has happened to poll() and to
> socket(), so it could happen again. Suppose the malloc() implementor
> assumes that the memory he gets from sbrk() is contiguous. Kaboom.*

poll() does not call malloc in libc or libc_r. Socket only calls malloc() in libc_r. libthr does not call malloc in either case. I'm not sure about libkse. Could you clarify this point? I agree, I wouldn't use sbrk() either. I'd use malloc(). If you have a small number of fixed allocations declaring them in the bss for maximum compaction is an alternative as well. No fragmentation, no allocator overhead, etc.

> *(If the sbrk() manual page said that this was the malloc() implementor's
> fault, maybe I'd risk it. But the manual page clearly says I shouldn't.
> ``Duh, why would anyone want to use more than one allocator library?'')*

I'm not really sure why they would. It seems like it would lead to increased inefficiencies and fragmentation. In the kernel we keep moving towards fewer allocators not more for this exact reason.

> *Even if I get malloc() under control somehow, I'll still have
30-odd
> unexplained VM pages going down the tubes. Is this, in fact, the stack?
> Why doesn't it start at 4K?*
>

It does start at 4k of allocated pages. I think I discussed this in my other email. With regards to the other pages that have been allocated to this process.. I remembered this today while thinking about another

freebsd-performance: Re: ten thousand small processes

problem. FreeBSD does defaults to map pages that are already available into an address space. This significantly reduces the number of faults in many applications due to spacial locality. This probably accounts for the memory used that has not actually been referenced.

FreeBSD and ELF just aren't well suited to extremely tiny binaries. I hope this answers your questions.

> ---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"
>

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"