

Re: ten thousand small processes

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2003-06/0086.html>

From: Jeff Roberson (jroberson_at_chesapeake.net)

Date: 06/24/03

Date: Mon, 23 Jun 2003 19:29:11 -0400 (EDT)

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

On Mon, 23 Jun 2003, Jeff Roberson wrote:

> *On 21 Jun 2003, D. J. Bernstein wrote:*

>

>> *FreeBSD 4.8. Test program: malloc(360); malloc(80); malloc(180);*

>> *malloc(16); malloc(440); sleep(10); _exit(0). Compile statically.*

>>

>> *The program ends up with 44KB RSS. Where is all that DRAM going? The*

>> *program also ends up with 168KB VSZ. Where is all that VM going?*

>>

>> *I don't care much about the 3-page text segment. But I do care about the*

>> *39 extra pages of VM, and the 8 extra pages of DRAM. There's no obstacle*

>> *to having a small program fit into one page per process; two or three*

>> *can be excused, but 39 is absurd. (Yes, I know that Solaris is worse.)*

>

The following should read "you need at least three pages"

> *Even small programs need page tables. On x86 unix you need at least*
> *pages for page tables for any process, if I'm counting correctly. One for*
> *the page directory, one page table for text, data, bss, and the guard*
> *page, and one page table for stack.*

>

> *32 of those 'pages of VM' are your initial stack size. They don't really*
> *consume any resources other than preventing anyone else from allocating*
> *overlapping pages. It's just the initial upper limit on the stack map*
> *which is allowed to grow. I haven't looked closely enough to find out*
> *what the other 7 might be.*

>

> *There is an obstacle to having a small program fit into one page.*
> *Actually, a significant one. First of all, you need protections on*
> *different sections of the actual executable image. Text must be read only*
> *since it is shared. Data is read write and bss is read-write. BSS is a*
> *pseudo section and not actually mapped from the file. Text and data both*
> *can be paged in from the binary in a demand paged system such as freebsd.*
> *Data can not be written out to its backing object and neither can text.*
> *Text can be shared while data changes are private and so the two must be*

freebsd-performance: Re: ten thousand small processes

> placed in separate pages. This topic is explored quite well in any modern
> operating systems book. I suggest you pick up "The Design and
> Implementation of the 4.4BSD Operating System". It is a little out dated
> but provides a good introduction to these topics. I really didn't do them
> justice with this paragraph.
>
> If demand paging, shared libraries, and the like are not suited for your
> problem perhaps you should look at an embedded operating system? Or DOS
> even.
>
> Cheers,
> Jeff
>
>
>> At least 2 pages appear to be wasted by exit(), because it brings in a
>> chunk of stdio, which uses 84 bytes of data and 316 bytes of bss. The
>> libc implementors clearly don't care about 316 bytes of memory, so why
>> don't they make those 316 bytes static? Why doesn't the compiler
>> automatically merge some bss into data when that saves a page? Why can't
>> I omit exit(), manually or automatically, when it's unreachable?
>>
>> Furthermore, malloc() appears to chew up a whole new page of DRAM for
>> each allocation, plus another page---is this counted in VSZ?---for an
>> anonymous mmap. Would it really be that difficult to fit 1076 bytes of
>> requested memory into the 3000-odd bytes available at the end of bss?
>>
>> I sure hope that there's some better explanation for the remaining 32
>> pages than "Well, we decided to allocate 131072 bytes of memory for the
>> stack," especially when I'm hard-limiting the stack to 4K before exec.
>>
>> ---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"
>>

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"