

Expanding the kernel stack

By **Jonathan Corbet**
May 29, 2014

Every process in the system occupies a certain amount of memory just by existing. Though it may seem small, one of the more important pieces of memory required for each process is a place to put the kernel stack. Since every process could conceivably be running in the kernel at the same time, each must have its own kernel stack area. If there are a lot of processes in the system, the space taken for kernel stacks can add up; the fact that the stack must be physically contiguous can stress the memory management subsystem as well. These concerns have always provided a strong motivation to keep the size of the kernel stack small.

For most of the history of Linux, on most architectures, the kernel stack has been put into an 8KB allocation — two physical pages. As recently as 2008 some developers were [trying to shrink the stack to 4KB](#), but that effort eventually proved to be unrealistic. Modern kernels can end up creating surprisingly deep call chains that just do not fit into a 4KB stack.

Increasingly, it seems, those call chains don't even fit into an 8KB stack on x86-64 systems. Recently, Minchan Kim tracked down a crash that turned out to be a stack overflow; he [responded](#) by proposing that it was time to double the stack size on x86-64 to 16KB. Such proposals have seen resistance before, and that happened this time around as well; Alan Cox [argued](#) that the solution is to be found elsewhere. But he seems to be nearly alone in that point of view.

Dave Chinner often has to deal with stack overflow problems, since they often occur with the XFS filesystem, which happens to be a bit more stack-hungry than others. He was [quite supportive](#) of this change:

8k stacks were never large enough to fit the linux IO architecture on x86-64, but nobody outside filesystem and IO developers has been willing to accept that argument as valid, despite regular stack overruns and filesystem having to add workaround after workaround to prevent stack overruns.

Linus was unconvinced at the outset, and he [made it clear](#) that work on reducing the kernel's stack footprint needs to continue. But Linus, too, seems to have come around to the idea that playing "whack-a-stack" is not going to be enough to solve the problem in a reliable way:

[S]o while I am basically planning on applying that patch, I also want to make sure that we fix the problems we do see and not just paper them over. The 8kB stack has been somewhat restrictive and painful for a while, and I'm ok with admitting that it is just getting too damn painful, but I don't want to just give up entirely when we have a known deep stack case.

Linus has also, unsurprisingly, made it clear that he is not interested in changing the stack size in the 3.15 kernel. But the 3.16 merge window can be expected to open in the near future; at that point, we may well see this patch go in as one of the first changes.

Index entries for this article

[Kernel](#)

[Kernel stack](#)

([Log in](#) to post comments)