

# Linus on Debuggers and Kernel Development

## Linus 谈调试器和内核如何发展

作者：Linus

翻译：Roman[AKA]

\*\*\*\*\*

译者序：关于 LINUX 内核的开发，我觉得这些观点都是正确的，因为观点都表达了不同的使用者的喜好。这些喜好都是需求。对于不同的使用者他们会更具自己的喜好，去使用不同的环境。这些都体现了 LINUX 的灵活性和可开发改造性，这些特点是别的系统所没有的。这就是我喜欢 LINUX 的一个原因。

关于不同的开发环境，个人有自己的喜爱和偏好，而各种环境都有自己的特点。我想这样的争论应该保持下去，因为这样的争论会促进 LINUX 的发展。能开发出各种各样优秀的工具。而这些工具的生存不在于一两个人，而是广大的 LINUX 的用户去决定的。

我想不同的观点将出现不同的风格。但是我不想 LINUX 的各种版本变的非常的大，而且他们相互都不互相融合。我认为 Linus 的观点是完全正确的，保证 LINUX 的内核的统一性和完整性。这样就保证了各种不同版本的 LINUX 一起发展。

\*\*\*\*\*

Sep 7, 2000, 14 :26 UTC (23 Talkback[s]) (3973 reads)

2000 年 9 月 7 日，14 : 26，联合技术公司 (23 对讲系统) (3973 读本)

From the Linux Kernel Mailing List:

来自 LINUX 内核邮件发送清单

Date: Wed, 6 Sep 2000 12:52:29 -0700 (PDT)

From: Linus Torvalds

Subject: Re: Availability of kdb

主题：KDB(内核编译器)的实用性/有效性

*On Wed, 6 Sep 2000, Tigran Aivazian wrote:*

*2000 年 9 月 6 日，周三，Tigran Aivazian 记录：*

*Very nice monologue, thanks. It would be great to know Linus' opinion. I mean, I knew Linus' opinion of some years' ago but perhaps it changed? He is a living being and not some set of rules written in stone so perhaps current stability/highquality of kdb suggests to Linus that it may be (just maybe) acceptable into official tree?*

*非常好的建议，谢谢，我很愿意了解 Linus 先生的观点。我在多年前就了解了 Linus 先生的观点，不知他的观点现在是不是改变了。他很有（不断创新的）生命力，而并不是一味地只知道奉行一些教条。因此，对于那些可接受的 KDB 方面的稳定性、高质量性建议，他很可能将其放入正式议事日程上去。*

I don't like debuggers. Never have, probably never will. I use gdb all the time, but I tend to use it not as a debugger, but as a disassembler on steroids that you can program.

我不喜欢调试器，从来不，大概永远不会喜欢。我只使用 GDB，而且我总是并不把它作为调试器来使用，只是将其作为一个可以用来分析程序的分解器来使用。

None of the arguments for a kernel debugger has touched me in the least. And trust me, over the years I've heard quite a lot of them. In the end, they tend to boil down to basically:

任何关于内核调试器的意见、争论都没有触动我，哪怕是丝毫。相信我，这么多年来我收到很多这方面的建议，到最后，他们都只能归结为很基础的（东西）：

-- It would be so much easier to do development, and we'd be able to add new things faster.

—— 开发应该变得更容易，我们能够更快的加入许多新的东西。

And quite frankly, I don't care. I don't think kernel development should be "easy". I do not condone single-stepping through code to find the bug. I do not think that extra visibility into the system is necessarily a good thing.

坦白地说，我并不在意（这个问题）。我认为对内核的开发不会是很容易的事。我不赞成那种通过一个个代码逐步去寻找错误的做法。我认为系统的额外可见度并不是一件必要的好事。

Apparently, if you follow the arguments, not having a kernel debugger leads to various maladies:

- you crash when something goes wrong, and you fsck and it takes forever and you get frustrated.
- people have given up on Linux kernel programming because it's too hard and too time-consuming
- - it takes longer to create new features.

很明显，如果你在没有使用一个内核调试器的情况下就附和这种观点：

—— 你会遇到一系列的问题：一旦出错，系统就会崩溃，你会失败；

—— Linux 内核编程太难太费时，人们会对其失去信心；

—— 创出新的特色需要一段很长的时间。

And nobody has explained to me why these are bad things.

没有一个人能向我解释这些问题。

To me, it's not a bug, it's a feature. Not only is it documented, but it's good, so it obviously cannot be a bug.

对我来说，这不是一个问题，这是它的特点。这不仅是已经有证明文件证明的，而且这是好事。因此很明显这不是一个问题。

"Takes longer to create new features" - this one in particular is not a very strong argument for having a debugger. It's not as if lack of features or new code would be a problem for Linux, or, in fact, for the software industry as a whole. Quite the reverse. My biggest job is to say "no" to new features, not trying to find them.

“创出新的特色需要一段很长的时间”——这一点在调试器方面尤为不是一个强有力的论据。对 Linux 来说，缺少特色或新代码不是一个问题，事实上，这对整个软件产业来说都是如此。相反，我的主要工作就是对那些新的特色/特征说“不”，而不是去寻找它们。

Oh. And sure, when things crash and you fck and you didn't even get a clue about what went wrong, you get frustrated. Tough. There are two kinds of reactions to that: you start being careful, or you start whining about a kernel debugger.

的确，当（系统）崩溃，你甚至不能获得一丝线索，只有失败，那么只能得到两种结果：你要么小心翼翼的重新开始；要么开始对内核调试器不断抱怨。

Quite frankly, I'd rather weed out the people who don't start being careful early rather than late. That sounds callous, and by God, it is callous. But it's not the kind of "if you can't stand the heat, get out the the kitchen" kind of remark that some people take it for. No, it's something much more deeper: I'd rather not work with people who aren't careful. It's darwinism in software development.

坦白的说，如果（工程进程中）出现粗心大意的情况，我宁愿摒弃那些在开始时就没有小心谨慎的人。这听上去很无情，就算是上帝听上去也会感觉无情。但这并不是人们所认为的那种“如果你不能承受压力，那就干脆离开”的情况。这里（所包含的意义）要更深一些。我宁可不和那些粗心大意的人一起工作。这就是软件发展的进化论。

It's a cold, callous argument that says that there are two kinds of people, and I'd rather not work with the second kind. Live with it.

这样把人分成两种是一个冷酷、无情的观点。我宁愿选择第一种人，忍受他们。

I'm a bastard. I have absolutely no clue why people can ever think otherwise. Yet they do. People think I'm a nice guy, and the fact is that I'm a scheming, conniving bastard who doesn't care for any hurt feelings or lost hours of work if it just results in what I consider to be a better system.

我是一个比较自私的人。我完全不知道人们为什么要从不同方面进行考虑，但是他们确实是（那么做的）。人们认为我是个好人的，但事实上我是个诡计多端的自私鬼，只要最终能得到我所认为的更好的系统，那么我对任何感情的伤害或工作时间的损失都不在乎。

And I'm not just saying that. I'm really not a very nice person. I can say "I don't care" with a straight face, and really mean it.

我并不只是（在口头上）说说而已，我真的不是一个很好的人。我能面无表情地说“我不在乎”，而且我确实不在乎。

I happen to believe that not having a kernel debugger forces people to think about their problem on a different level than with a debugger. I think that without a debugger, you don't get into that mindset where you know how it behaves, and then you fix it from there. Without a debugger, you tend to think about problems another way. You want to understand things on a different level.

我相信不使用内核调试器会迫使人们在一个不同的层次上考虑问题。我认为如果你不使用调试器，

你就不能得知他如何运转以及你如何处理，你就试图从别的角度去考虑问题。你会想在不同的层次上理解事情。

It's partly "source vs binary", but it's more than that. It's not that you have to look at the sources (of course you have to - and any good debugger will make that easy). It's that you have to look at the level above sources. At the meaning of things. Without a debugger, you basically have to go the next step: understand what the program does. Not just that particular line.

在一定程度上更多的是“源代码对二进制”（的问题）。你不必不得不去查看源代码（当然你可以去查看，任何优良的调试器使其轻而易举）。你必须在源代码之上的层次进行查看。就是说，不使用内核调试器的话，你将不得不去理解程序在做什么，而不仅仅是特定的（代码）行。

And quite frankly, for most of the real problems (as opposed to the stupid bugs - of which there are many, as the latest crap with "truncate()" has shown us) a debugger doesn't much help. And the real problems are what I worry about. The rest is just details. It will get fixed eventually.

坦白的说，对于许多实际问题（这和错误截然不同的，那些愚蠢的错误是那么多）来说，调试器并没有多大的作用。这些实际问题正是我所担心的。剩下的就是一些细节了，他们最终都会被确定下来。

I do realize that others disagree. And I'm not your Mom. You can use a kernel debugger if you want to, and I won't give you the cold shoulder because you have "sullied" yourself. But I'm not going to help you use one, and I would frankly prefer people not to use kernel debuggers that much. So I don't make it part of the standard distribution, and if the existing debuggers aren't very well known I won't shed a tear over it.

我能理解那些与我不一致的意见。我不是你们的母亲，如果你愿意的话你可以使用内核调试器，我不会因为你自己的“毁誉”而轻视你。但是我不会去协助你使用他，我真诚希望人们不要高频率地使用内核调试器。因此我不会将其作为评定的标准，如果现有的调试器没有被人们很好的了解，我不会去（刻意）糟蹋贬低他。

Because I'm a bastard, and proud of it!

因为我是一个比较自私的人，但是我以此为荣！

Linus



自由、协作、创造 — 为了明天  
“来自大雪山的大雁阿卡”

更多精彩文章，请访问：<http://www.AKA.org.cn> 精彩文章 栏目  
本文如有翻译错误或不妥，请 Email 至 [AKAMagazine@yahoo.com](mailto:AKAMagazine@yahoo.com)

---