

第 24 章 内存虚拟化总结对话

学生：（大口吸气）哇，这部分内容很多。

教授：是的，那么……

学生：那么，我应该如何记住这一切？你懂的，为了考试？

教授：天啊，我希望这不是你试图记住它的原因。

学生：那我为什么要记住呢？

教授：算了吧，我以为你领会更好。你试图在这里学习一些东西，这样当你走进这个世界时，就会明白系统是如何工作的。

学生：嗯……你能举个例子吗？

教授：当然！当我还在研究生院时，有一次我和朋友正在测量内存存取的时间，有时候这些数字比我们预期的要高。我们认为所有数据都很好地融入了二级硬件缓存中，你知道，因此应该非常快速地访问。

学生：（点头）

教授：我们无法弄清楚发生了什么事。那么你在这种情况下做什么？很容易，问一位教授！于是我们去问一位教授，他看过我们制作的图表，简单地说“TLB”。啊哈！当然，TLB 未命中！我们为什么没有想到这个？有一个好的虚拟内存模型可以帮助诊断各种有趣的性能问题。

学生：我想我明白了。我要尝试建立这些关于事情如何工作的心智模型，以便我在那里独立工作，当系统不像预期的那样行事时，不会感到惊讶。我甚至应该能够预测系统将如何工作，只要想想它就行。

教授：确实如此。那么你学到了什么？关于虚拟内存如何工作的，你的心智模型有哪些？

学生：我认为我现在对进程引用内存时会发生什么有了很好的概念，正如您多次说过的那样，每次获取指令以及显式加载和存储时都会发生。

教授：听起来不错，说下去。

学生：那么，我会永远记住的一件事是，我们在用户程序中看到的地址，例如用 C 语言编写的……

教授：还有什么其他的语言？

学生：（继续）……是的，我知道你喜欢 C，我也是！无论如何，正如我所说的，我现在真的知道，我们在程序中可以观察到的所有地址都是虚拟地址。作为一名程序员，我只是看到了数据和代码在内存中的假象。我曾经认为能够打印指针的地址是很酷的，但现在我发现它令人沮丧——它只是一个虚拟地址！我看不到数据所在真实物理地址。

教授：你看不到，操作系统肯定会向你隐藏的。还有什么？

学生：嗯，我认为 TLB 是一个非常关键的部分，为系统提供了一个地址转换的小硬件缓存。页表通常相当大，因此放在大而慢的内存中。没有 TLB，程序运行速度肯定会慢得

多。TLB 似乎真的让虚拟内存成为可能。我无法想象构建一个没有 TLB 的系统！我想到了一个超出 TLB 覆盖范围的程序：所有那些 TLB 未命中，简直不敢看。

教授：是的，蒙住孩子们的眼睛！除了 TLB，你还学到了什么？

学生：我现在也明白，页表是需要了解的数据结构之一。它只是一个数据结构，这意味着几乎可以使用任何结构。我们从简单的结构（如数组，即线性页表）开始，一直到多级表（它们看起来像树），甚至像内核虚拟内存中的可分页页表一样疯狂。全是为了在内存中节省一点空间！

教授：的确如此。

学生：还有一件更重要的事情：我了解到，地址转换结构需要足够灵活，以支持程序员想要处理的地址空间。在这个意义上，像多级表这样的结构是完美的。它们只在用户需要一部分地址空间时才创建表空间，因此几乎没有浪费。早期的尝试，比如简单的基址和界限寄存器，只是不够灵活。这些结构需要与用户期望和想要的虚拟内存系统相匹配。

教授：这是一个很好的观点。我们所学到的关于交换到磁盘的所有内容的情况如何？

学生：好的，学习肯定很有趣，而且很好地知道页替换的工作原理。一些基本的策略是很明显的（比如 LRU），但是建立一个真正的虚拟内存系统似乎更有趣，就像我们在 VMS 案例研究中看到的一样。但不知何故，我发现这些机制更有趣，而策略则不太有趣。

教授：哦，那是为什么？

学生：正如你所说的那样，最终解决策略问题的好办法很简单：购买更多的内存。但是你需要理解的机制才能知道事情是如何运作的。说到……

教授：什么？

学生：好吧，我的机器现在运行速度有点慢……而且内存肯定不会太贵……

教授：噢，很好，很好！这里有些钱，去买一些 DRAM，小事情。

学生：谢谢教授！我再也不会交换到硬盘了——或者，如果发生交换，至少我会知道实际发生了什么！