

13.6.1 虚拟化

虚拟化重回时代。它第一次出现在1967年的IBM CP/CMS系统中，现在它重回奔腾平台。最近许多计算机在裸机上运行管理程序，如图13-12所示。管理程序会创建多个虚拟机，每个虚拟机有单独的操作系统。有些计算机利用一个虚拟机为遗产应用创建Windows系统，利用几个虚拟机为当前应用运行Linux系统，或许也会在其他虚拟机上运行若干实验性操作系统。这种现象在第8章已经讨论，并且是未来的发展趋势。

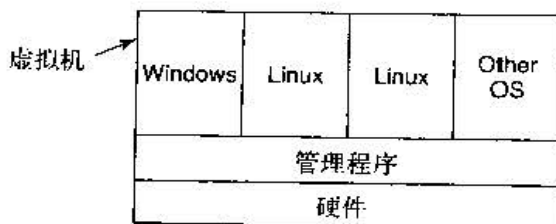


图13-12 运行4个虚拟机的管理程序

13.6.2 多核芯片

多核芯片已经出现，但即使是双核，针对它们的操作系统还没有很好地利用其能力，更不用提64核。这些核会做什么事情？它们需要哪些软件？这些目前都是未知的。起初，人们试图通过对当前操作系统打补丁的方法来支持多核，但锁表问题和其他软件资源的问题使得这种方法不太可能成功，因此需要全新的思路来解决这些问题。

虚拟化和多核芯片的结合创造了一个全新的环境，这里CPU的数目是可编程的。对于8核芯片，软件可以在下列情况做同样的事情：只利用1个CPU而忽略其他7个；使用全部8个CPU；利用双道虚拟化获得16个虚拟CPU；利用四道虚拟化获得32个虚拟CPU；或更多其他组合。程序可以在启动时指定所需CPU数目，由操作系统来保证程序需求的满足。

13.6.3 大型地址空间操作系统

随着计算机从32位地址空间转向64位地址空间，操作系统设计中的重大转变成为可能。32位地址空间并不大。如果你通过给地球上的每个人提供他或她自己的字节来试图分割 2^{32} 个字节，那么将没有足够的字节可以提供。相反， 2^{64} 大约是 2×10^{19} 。现在每个人可以得到他或她个人的3GB大的一块。

对于 2×10^{19} 字节的地址空间我们能做什么呢？首先，可以淘汰文件系统概念。作为替代，所有文件在概念上可以始终保存在（虚拟）内存中。毕竟在那里存在足够的空间，可以放下超过10亿部全长的电影，每一部压缩到4GB。

另一个可能的用途是永久对象存储。对象可以在地址空间中创建，并且保存在那里直到所有对它们的引用消失，在此时它们可以自动被删除。这样的对象在地址空间中是永久的，甚至是在关机和重新启动计算机的时候。有了64位的地址空间，在用光地址空间之前，可以用每秒100MB的速率创建对象长达5000年。当然，为了实际存储这么大量的数据，需要许多磁盘存储器用于分页交换，但是在历史上这是第一次限制因素是磁盘，而不是地址空间。

由于大量数目的对象在地址空间中，允许多个进程同时在相同的地址空间中运行，以便以一般的方式共享对象就变得十分有趣了。这样的设计显然会通向与我们现在所使用的操作系统完全不同的操作系统。有关这一概念的某些思想包含在参考文献（Chase等人，1994）中。

就64位地址而言，另一个必须重新思考的操作系统问题是虚拟内存。对于 2^{64} 字节的虚拟地址空间和8KB的页面，我们有 2^{51} 个页面。常规的页表不能很好地按比例变换到这样的大小，所以需要别的东西。反转的页表是可行的，但是也有人提出了其他的想法（Talluri等人，1995）。无论如何，64位操作系统为新的研究提供了大量的余地。

13.6.4 联网

当前的操作系统是为独立的计算机而设计的。联网是事后添加的，并且一般通过特殊的程序访问，例如Web浏览器、FTP或telnet。在将来，联网或许将会是所有操作系统的基础。不具备网络连接的独立的计算机就像是没有连接到电话网的电话机一样罕见。并且很可能几Gbps的连接是标准的速率。

操作系统将不得不改变以适应这一范型的转变。本地数据与远程数据的区别可能会模糊到这样的程度：实际上没有人知道或者关心数据存放在什么地方。任何地方的计算机能够像本地数据一样处理任何地方的数据。在一个有限的范围内，对于NFS而言这已经成为现实，但是它很可能变得更加普遍并且更好地集成。