

速缓存中的情况下,当其中某个CPU修改了该字,所有其他高速缓存中的该字都会被自动地并且原子性地删除来确保一致性。这个过程称为窥探(snooping)。

这样设计的结果是多核芯片就相当于小的多处理机。实际上,多核芯片时常被称为片级多处理机(Chip-level MultiProcessors, CMP)。从软件的角度来看, CMP与基于总线或多处理机和使用交换网络的多处理机并没有太大的差别。不过,它们还是存在着若干的差别。例如,对基于总线或多处理机,每个CPU拥有自己的高速缓存,如图8-2b以及图1-8b的AMD设计所示。在图1-8a所示的Intel使用的共享高速缓存的设计并没有出现在其他的多处理机中。共享二级高速缓存会影响性能。如果一个核需要很多高速缓存空间,而另一个核不需要,这样的设计允许它们各自使用所需的高速缓存。但另一方面,共享高速缓存也让一个贪婪的核损害其他核的性能成为了可能。

CMP与其他更大的多处理机之间的另一个差异是容错。因为CPU之间的连接非常紧密,一个共享模块的失效可能导致许多CPU同时出错。而这样的情况在传统的多处理机中是很少出现的。

除了所有核都是对等的对称多核芯片之外,还有一类多核芯片被称为片上系统(system on a chip)。这些芯片含有一个或者多个主CPU,但是同时还包含若干个专用核,例如视频与音频解码器、加密芯片、网络接口等。这些核共同构成了完整的片上计算机系统。

正如过去已经发生的,硬件的发展常常领先于软件。多核的时代已经来临,但是我们还不具备为它们编写应用程序的能力。现有的编程语言并不适合编写高度并行的代码,同时适用的编译器和调试工具还很匮乏。几乎没有几个程序员有编写并行程序的经验,而大部分程序员对于如何将工作划分为若干可以并行执行的块(package)知之甚少。同步、消除竞争、避免死锁成为了程序员的噩梦,同时也影响到了性能。信号量(semaphore)并不能解决问题。除了这些问题,什么样的应用真的需要使用数百个核尚不明确。自然语言语音识别可能需要大量的计算能力,但这里的问题并不是缺少时钟周期,而是缺少可行的算法。简而言之,或许硬件开发人员正在发布软件开发人员不知道如何使用而用户也并不需要的产品。

### 8.1.2 多处理机操作系统类型

让我们从对多处理机硬件的讨论转到多处理机软件,特别是多处理机操作系统上来。这里有各种可能的方法。接下来将讨论其中的三种。需要强调的是所有这些方法除了适用于多核系统之外,同样适用于包含多个分离CPU的系统。

#### 1. 每个CPU有自己的操作系统

组织一个多处理机操作系统的可能的最简单的方法是,静态地把存储器划分成和CPU一样多的各个部分,为每个CPU提供其私有存储器以及操作系统的各自私有副本。实际上 $n$ 个CPU以 $n$ 个独立计算机的形式运行。这样做一个明显的优点是,允许所有的CPU共享操作系统的代码,而且只需要提供数据的私有副本,如图8-7所示。

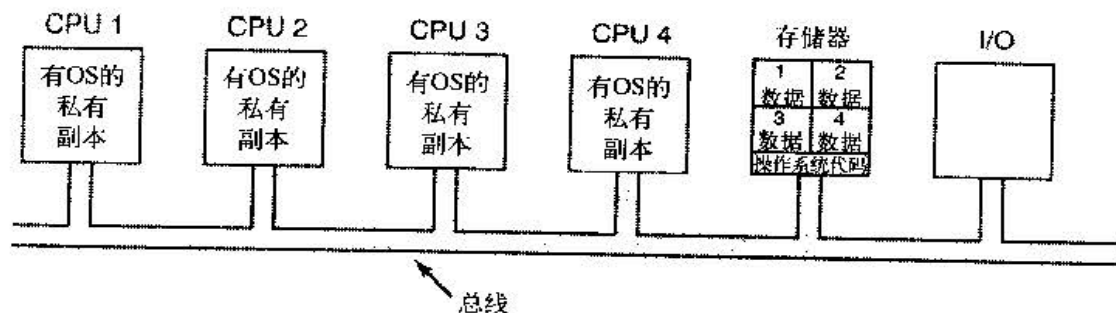


图8-7 在4个CPU中划分多处理机存储器,但共享一个操作系统代码的副本。

标有“数据”字样的方框是每个CPU的操作系统私有数据

这一机制比有 $n$ 个分离的计算机要好,因为它允许所有的机器共享一套磁盘及其他的I/O设备,它还允许灵活地共享存储器。例如,即便使用静态内存分配,一个CPU也可以获得极大的一块内存,从而高效地执行代码。另外,由于生产者能够直接把数据写入存储器,从而使得消费者从生产者写入的位置取出数据,因此进程之间可以高效地通信。况且,从操作系统的角度看,每个CPU都有自己的操作系统非常自然。