CPU保持空闲，直到对应的时间片结束为止。

有关群调度是如何工作的例子在图8-15中给出。图8-15中有一台带6个CPU的多处理机，由5个进程A到E使用，总共有24个就绪线程。在时间槽（time slot）0，线程A_0至A_6被调度运行。在时间槽1，调度线程B_0、B_1、B_2、C_0、C_1和C_2被调度运行。在时间槽2，进程D的5个线程以及E_0运行。剩下的6个线程属于E，在时间槽3中运行。然后周期重复进行，时间槽4与时间槽0一样，以此类推。

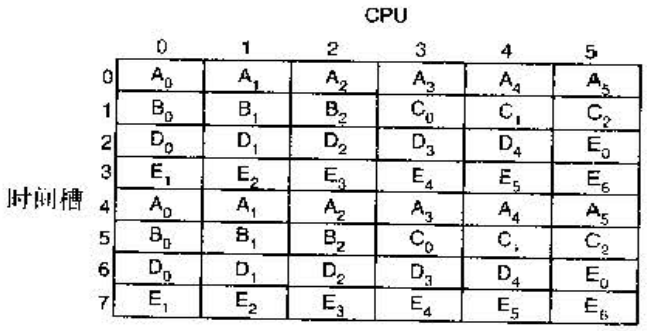

图8-15 群调度

群调度的思想是，让一个进程的所有线程一起运行，这样，如果其中一个线程向另一个线程发送请求，接受方几乎会立即得到消息，并且几乎能够立即应答。在图8-15中，由于进程的所有线程在同一个时间片内一起运行，它们可以在一个时间片内发送和接受大量的消息，从而消除了图8-14中的问题。

8.2 多计算机

多处理机流行和有吸引力的原因是，它们提供了一个简单的通信模型：所有CPU共享一个公用存储器。进程可以向存储器写消息，然后被其他进程读取。可以使用互斥信号量、信号量、管程（monitor）和其他适合的技术实现同步。惟一美中不足的是，大型多处理机构造困难，因而造价高昂。

为了解决这个问题，人们在多计算机（multicomputers）领域中进行了很多研究。多计算机是紧耦合CPU，不共享存储器。每台计算机有自己的存储器，如图8-1b所示。众所周知，这些系统有各种其他的名称，如机群计算机（cluster computers）以及工作站机群（Clusters of Workstations，COWS）。

多计算机容易构造，因为其基本部件只是一台配有高性能网络接口卡的PC裸机。当然，获得高性能的秘密是巧妙地设计互连网络以及接口卡。这个问题与在一台多处理机中构造共享存储器是完全类似的。但是，由于目标是在微秒（microsecond）数量级上发送消息，而不是在纳秒（nanosecond）数量级上访问存储器，所以这是一个相对简单、便宜且容易实现的任务。

在下面几节中，我们将首先简要地介绍多计算机硬件，特别是互连硬件。然后，我们将讨论软件，从低层通信软件开始，接着是高层通信软件。我们还将讨论在没有共享存储器的系统中实现共享存储器的方法。最后，我们将讨论调度和负载平衡的问题。

8.2.1 多计算机硬件

一台多计算机的基本节点包括一个CPU、存储器、一个网络接口，有时还有一个硬盘。节点可以封装在标准的PC机箱中，不过通常没有图像适配卡、显示器、键盘和鼠标等。在某些情况下，PC机中有一块2通道或4通道的多处理机主板，可能带有双核或者四核芯片而不是单个CPU，不过为了简化问题，我们假设每个节点有一个CPU。通常成百个甚至上千个节点连接在一起组成一个多计算机。下面我们将介绍一些关于硬件如何组织的内容。

1. 互连技术

在每个节点上有一块网卡，带有一根或两根从网卡上接出的电缆（或光纤）。这些电缆或者连到其他的节点上，或者连到交换机上。在小型系统中，可能会有一个按照图8-16a的星型拓扑结构连接所有节点的的交换机。现代交换型以太网就采用了这种拓扑结构。

作为单一交换机设计的另一种选择，节点可以组成一个环，有两根线从网络接口卡上出来，一根去连接左面的节点，另一根去连接右面的节点，如图8-16b所示。在这种拓扑结构中不需要交换机，所以图中也没有。

图8-16c中的网格（grid或mesh）是一种在许多商业系统中应用的二维设计。它相当规整，而且容易扩展为大规模系统。这种系统有一个直径（diameter），即在任意两个节点之间的最长路径，并且该值只按照节点数目的平方根增加。网格的变种是双卷面（double torus），如图8-16d所示，这是一种边连通