下也是如此。这一思想起因于（Aron和Druschel，1999）。关于更详细的细节，请参阅他们的论文。

一般而言，有两种方法管理I/O：中断和轮询。中断具有较低的等待时间，也就是说，它们在事件本身之后立即发生，具有很少的延迟或者没有延迟。另一方面，对于现代CPU而言，由于需要上下文切换以及对于流水线、TLB和高速缓存的影响，中断具有相当大的开销。

替代中断的是让应用程序对它本身期待的事件进行轮询。这样做避免了中断，但是可能存在相当长的等待时间，因为一个事件可能正好发生在一次轮询之后，在这种情况下它就要等待几乎整个轮询间隔。平均而言，等待时间是轮询间隔的一半。

对于某些应用而言，中断的开销和轮询的等待时间都是不能接受的。例如，考虑一个高性能的网络，如千兆位以太网。该网络能够每12μs接收或者发送一个全长的数据包。为了以优化的输出性能运行，每隔12μs就应该发出一个数据包。

达到这一速率的一种方法是当一个数据包传输完成时引发一个中断，或者将辅助定时器设置为每12μs中断一次。问题是在一个300 MHz的Pentium II计算机上该中断经实测要花费4.45μs的时间（Aron和Druschel，1999）。这样的开销比20世纪70年代的计算机好不了多少。例如，在大多数小型机上，一个中断要占用4个总线周期：将程序计数器和PSW压入堆栈并且加载一个新的程序计数器和PSW。现如今涉及流水线、MMU、TLB和高速缓存，更是增加了大量的开销。这些影响可能在时间上使情况变得更坏而不是变得更好，因此抵消了更快的时钟速率。

软定时器（soft timer）避免了中断。无论何时当内核因某种其他原因在运行时，在它返回到用户态之前，它都要检查实时时钟以了解软定时器是否到期。如果这个定时器已经到期，则执行被调度的事件（例如，传送数据包或者检查到来的数据包），而无需切换到内核态，因为系统已经在内核态。在完成工作之后，软定时器被复位以便再次闹响。要做的全部工作是将当前时钟值复制给定时器并且将超时间隔加上。

软定时器随着因为其他原因进入内核的频率而脉动。这些原因包括：

1) 系统调用。

2) TLB未命中。

3) 页面故障。

4) I/O中断。

5) CPU变成空闲。

为了了解这些事件发生得有多频繁，Aron和Druschel对于几种CPU负载进行了测量，包括全负载Web服务器、具有计算约束后台作业的Web服务器、从因特网上播放实时音频以及重编译UNIX内核。进入内核的平均进入率在2μs到1μs之间变化，其中大约一半是系统调用。因此，对于一阶近似，让一个软定时器每隔2μs闹响一次是可行的，虽然这样做偶尔会错过最终时限。对于发送数据包或者轮询到来的数据包这样的应用而言，有时可能晚10μs比让中断消耗35%的CPU时间要好。

当然，可能有一段时间不存在系统调用、TLB未命中或页面故障，在这些情况下，没有软定时器会闹响。为了在这些时间间隔上设置一个最大值，可以将辅助硬件定时器设置为每隔一定时间（例如1ms）闹响一次。如果应用程序对于偶然的时间间隔能够忍受每秒只有1000个数据包，那么软定时器和低频硬件定时器的组合可能比纯粹的中断驱动I/O或者纯粹的轮询要好。

5.6 用户界面：键盘、鼠标和监视器

每台通用计算机都配有一个键盘和一个监视器（并且通常还有一只鼠标），使人们可以与之交互。尽管键盘和监视器在技术上是独立的设备，但是它们紧密地一同工作。在大型机上，通常存在许多远程用户，每个用户拥有一个设备，该设备包括一个键盘和一个连在一起的显示器作为一个单位。这些设备在历史上被称为终端（terminal）。人们通常继续使用该术语，即便是讨论个人计算机时（主要是因为缺乏更好的术语）。

5.6.1 输入软件

用户输入主要来自键盘和鼠标，所以我们要了解它们。在个人计算机上，键盘包含一个嵌入式微处理器，该微处理器通过一个特殊的串行端口与主板上的控制芯片通信（尽管键盘越来越多地连接到USB端口上）。每当一个键被按下的时候都会产生一个中断，并且每当一个键被释放的时候还会产生第二个