

I/O性能问题

- ⊙ 使**CPU**利用率尽可能不被**I/O**降低
- ⊙ 使**CPU**尽可能摆脱**I/O**
 - 减少或缓解速度差距 → 缓冲技术
 - 使**CPU**不等待**I/O** → 异步**I/O**
 - 让**CPU**摆脱**I/O**操作 → **DMA**、通道

，由 I/O 部件来完成 I/O 的具体过程。



异步传输

- Windows提供两种模式的I/O操作：
异步和同步
- 异步模式：用于优化应用程序的性能
 - 通过异步I/O，应用程序可以启动一个I/O操作，然后在I/O请求执行的同时继续处理
 - 基本思想：填充I/O操作间等待的CPU时间
- 同步I/O：应用程序被阻塞直到I/O操作完成

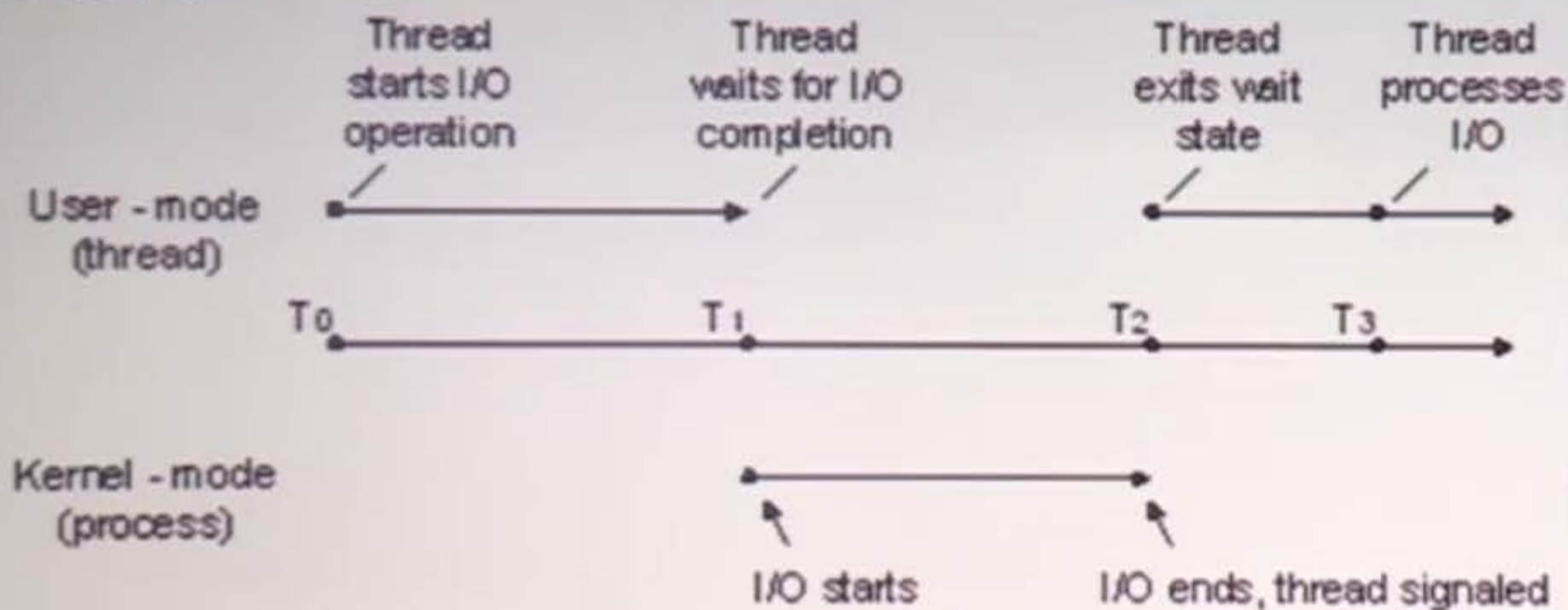
请求之后，然后呢阻塞直到这个 I/O 操作完成。



同步传输I/O流程

- 在I/O处理过程中，CPU处于空闲等待状态
- 而在处理数据的过程中，不能同时进行I/O操作

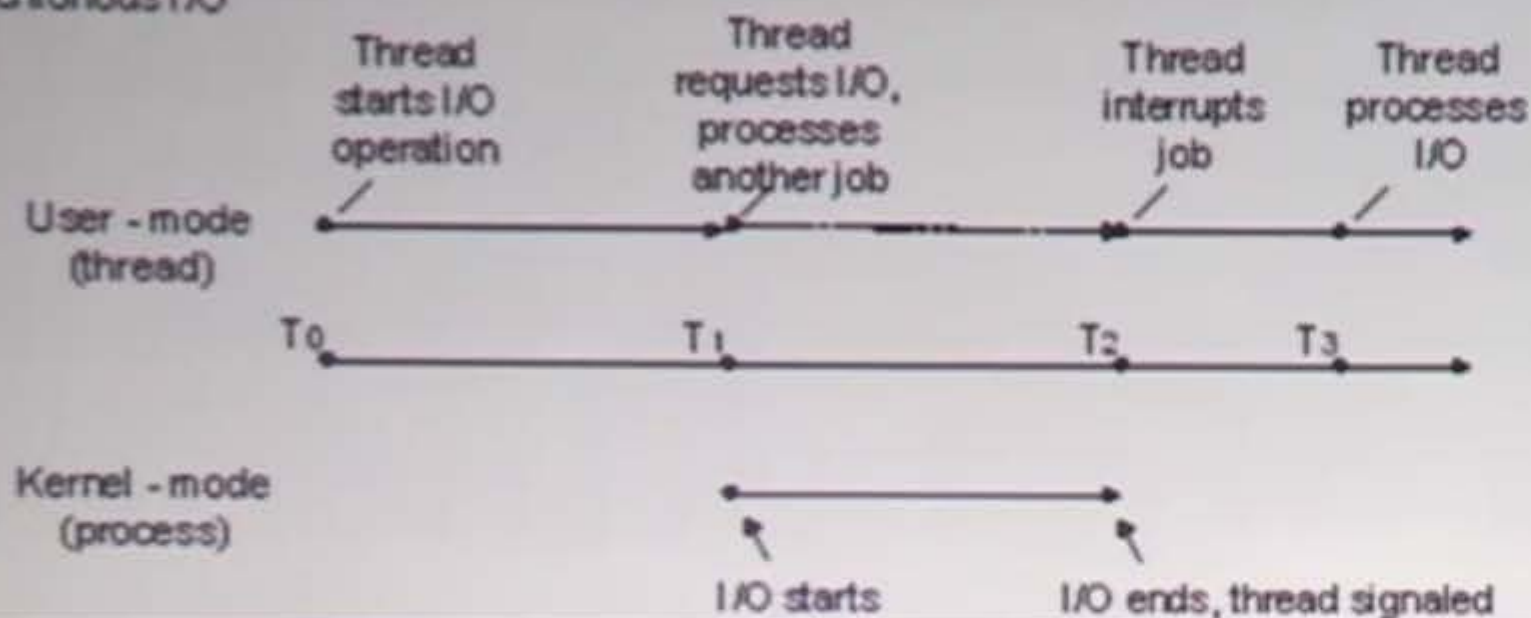
Synchronous I/O



直到 I/O 请求完成了，它才能够对 I/O 请求的结果做相应的处理。

异步传输I/O的基本思想

Asynchronous I/O



系统实现

- 通过切换到其他线程保证CPU利用率
- 对少量数据的I/O操作会引入切换的开销

用户实现

- 将访问控制分成两段进行
- 发出读取指令后继续做其他操作
- 当需要用读入的数据的时候，再使用wait命令等待其完成
- 不引入线程切换，减少开销



本讲重点

- ◎ 掌握I/O系统的概念
 - 理解I/O设备的特点及分类
 - 理解I/O管理的任务
- ◎ 掌握I/O管理的解决方案
 - 了解I/O硬件组成
 - 理解I/O控制方式
 - 掌握I/O软件层次及功能
- ◎ 掌握I/O相关技术
- ◎ 了解I/O性能提高的各种解决方案



本周要求

重点阅读教材

第5章相关内容：5.1、5.2、5.4

重点概念

I/O管理的各种任务 设备分类

设备独立性(设备无关性)

I/O控制方式

I/O硬件工作原理 I/O软件层次

缓冲技术 独占设备分配 共享设备分配

设备驱动程序



下面我们简单讨论一下 I/O 性能问题。解决 I/O 性能呢从两个方面入手，第一个方面呢是使 CPU 利用率尽可能不被 I/O 降低。第二个呢是使得 CPU 尽可能的摆脱 I/O。由于 CPU 的处理速度和 I/O 处理速度 差距很大，所以可以通过缓冲技术 来减少或者缓解它们的速度差异。这样就使得 CPU 的利用率不会被 I/O 降低很多。如何使 CPU 不等待 I/O 呢？那么可以采用异步 I/O 技术。如何让 CPU 摆脱 I/O 操作呢？那就要运用一些 I/O 部件 通过 DMA、通道这样的 I/O 部件 可以使得 CPU 摆脱了 I/O，由 I/O 部件来完成 I/O 的具体过程。下面我们介绍一下异步 I/O，Windows 提供了 两种模式的 I/O 操作：一种是异步，一种是同步。异步模式呢，主要是用于优化应用程序的性能，通过异步 I/O，应用程序可以启动一个 I/O 操作，然后在这个 I/O 操作完成的过程当中，继续去处理其它的一些操作。它的基本思想是用其它的一些操作来填充 等待 I/O 操作的这样的一种 CPU 时间。而同步 I/O 呢指的是应用程序提出了一个 I/O 请求之后，然后呢阻塞直到这个 I/O 操作完成。我们来看一下同步传输 I/O 的流程。也就是在 I/O 处理过程当中，CPU 呢实际上是处于空闲等待状态，而在处理数据的过程中呢，也不能同时进行 I/O 操作。我们这张图其实示意了说，某一个线程 如果要进行 I/O，提出了 I/O 请求，那么它在做一些 I/O 请求准备工作的时候呢，它在 CPU 上执行，那么当提出了 I/O 请求 由操作系统完成这个 I/O 请求的过程的过程中呢，这个线程就停在那里了，所以它们不能做任何事情。直到 I/O 请求完成了，它能够对 I/O 请求的结果做相应的处理。下面我们来看一下 异步传输 I/O 的基本思想，当用户进程提出 I/O 请求之后，作为操作系统，它可以通过 切换到其他的线程来保障 CPU 的利用率，但是 线程的切换会带来开销。如果我本次 I/O 请求只是少量 的数据，很快就能完成，那么这个线程切换的开销就不划算。那么因此用户其实可以通过异步 I/O 的这种模式来完成 I/O 操作。具体来讲，在用户实现的时候，把访问控制分成两阶段来进行，首先它提出了一个读取请求之后，转去做一些和读取请求无关的一些 其他的操作，也就是用与读写请求无关的这样的一些操作来 填充等待的时间，它不需要等待了，可以继续执行其它的操作。当做完那些操作之后，返回来第二阶段，那么它如果还需要对数据进行处理 而数据没有到达的时候，再去 使用 wait 命令来等待结果的完成。但是如果它进入第二阶段 而由于 I/O 的数量比较少，很快就完成了。所以这个进程进入第二阶段的时候呢，数据已经准备就绪，它就可以进行后续的处理工作了。也就不需要再等待，不需要再切换到其它线程了，这样就提高了 I/O 的一个性能。本讲的重点

呢，是要掌握 I/O 系统的概念，通过设备的特点，设备的分类以及 I/O 管理完成什么样的任务，了解整个操作系统当中 I/O 管理这一部分它的主要的工作。另外呢，希望大家掌握 I/O 管理的各种解决方案。比如说，你要了解硬件的组成。要理解 I/O 控制方式。要掌握在实现 I/O 软件的时候分层的设计思想。要掌握缓冲技术这样的 I/O 相关技术。要了解提高 I/O 性能的各种解决方案。关于教材的阅读呢，我们主要是第 5 章的部分章节。重点的概念呢，我们这里也给出了，好，本讲的内容呢就介绍到这里。谢谢大家！