第10章 同步设备I/O与异步设备I/O

10.4.2 异步设备I/O的注意事项

P294. 异步设备I/O的注意事项

设备驱动程序不必以先入先出的方式来处理队列中的I/O请求，也就是说，最先投递的异步I/O请求可能不会最先进行处理

在异步I/O请求完成之前，一定不能移动或是销毁在投递I/O请求时所使用的数据缓冲区和OVERLAPPED结构（注意，一个OVERLAPPED对象应该只与一个异步I/O请求相关联）

10.5 接收I/O请求完成通知

|  |  |
| --- | --- |
| 技 术 | 摘 要 |
| 触发设备内核对象 | 当向一个设备投递多个I/O请求的时候，这种方法没有什么用。因为无法确定设备内核对象的触发是由于哪个I/O请求处理完成导致的。它允许一个线程发出I/O请求，另一个线程对结果进行处理。 |
| 触发事件内核对象 | 这种方法允许我们向设备发出多个I/O请求。它允许一个线程发出I/O请求，另一个线程对结果进行处理。 |
| 使用可提醒I/O | 这种方法允许我们向设备发出多个I/O请求。投递I/O请求的线程必须对结果进行处理。 |
| 使用I/O完成端口 | 这种方法允许我们向设备发出多个I/O请求。它允许一个线程发出I/O请求，另一个线程对结果进行处理。这项技术具有高度的伸缩性和最佳的灵活性。 |

10.5.4 I/O完成端口（P306）

P307. IOCP背后的理论是并发运行的线程的数量必须有一个上限。

如果可运行的线程数太多，CPU会浪费大量的时间片在线程之间切换，每个线程执行任务的时间也会减少，导致线程执行效率的下降。合适的可运行线程数应该等于CPU的核心数目。

P307. 线程池

频繁的创建线程是有开销的（时间和空间），在应用程序启动的时候预先创建一个线程池，并使线程池中的线程在应用程序执行过程中随时保持可用，这种“饿汉式”加载的策略，能有效降低频繁创建线程的系统开销。

P309. 与IOCP内核对象相关联的五个数据结构

**设备列表**：表示与该IOCP相关联的一个或多个设备

每条记录包含

|  |  |
| --- | --- |
| hDevice | hCompletionKey |

当满足以下条件时，会在列表中添加新项：

（1）CreateIoCompletionPort被调用

当满足以下条件时，会在列表中删除一项：

（1）设备句柄被关闭

**I/O完成队列**（FIFO）：取出I/O完成队列中的I/O完成包是以先入先出的方式来进行的

|  |  |  |  |
| --- | --- | --- | --- |
| dwBytesTransferred | dwCompletionKey | pOverlapped | dwError |

当满足以下条件时，会在列表中添加新项：

（1）I/O请求完成（注意，如果一个设备有完成端口与之相关联，那么当我们向它投递一个异步I/O请求时，Windows会将此异步操作的结果添加在完成端口的I/O完成队列中，即使异步请求是以同步方式完成的。P312）

（2）PostQueuedCompletionStatus被调用

当满足以下条件时，会在列表中删除一项：

（1）完成端口从等待线程队列中删除一项（与IOCP相关联的线程池中的线程为了等待I/O完成队列出现I/O完成包而处理睡眠状态，一旦发现I/O完成队列中出现I/O完成包，线程就会立即取出此I/O完成包，并切换到可执行状态进行处理）

**等待线程队列**（LIFO）：唤醒那些调用了GQCS的线程是以后入先出的方式来进行的

|  |
| --- |
| dwThreadID |

当满足以下条件时，会在列表中添加新项：

（1）线程调用GetQueuedCompletionStatue

当满足以下条件时，会在列表中删除一项：

（2）I/O完成队列不为空，并且正在运行的线程数小于最大并发线程数（GQCS会从I/O完成队列中取出I/O完成包，接着将dwThreadID转移到已释放线程列表，最后GQCS函数返回，线程得以继续执行）

**已释放线程列表**

|  |
| --- |
| dwThreadID |

当满足以下条件时，会在列表中添加新项：

（1）完成端口在等待线程队列中唤醒了一个线程（必须满足两个条件：I/O完成队列中存在I/O完成包，目前正在运行的线程数少于最大并发线程数）

（2）已暂停的线程被唤醒

当满足以下条件时，会在列表中删除一项：

（1）线程再次调用GQCS（dwThreadID再次回到等待线程队列）

（2）线程调用一个函数将自己挂起（dwThreadID转移到已暂停线程列表）

**已暂停线程列表**

|  |
| --- |
| dwThreadID |

当满足以下条件时，会在列表中添加新项：

（1）已释放的线程调用一个函数将自己挂起

当满足以下条件时，会在列表中删除一项：

（1）已挂起的线程被唤醒（dwThreadID回到已释放线程列表）

P311. GetQueuedCompletionStatus的基本任务

将调用线程切换到睡眠状态，直到指定的IOCP的I/O完成队列中出现一项I/O完成包，或者线程等待的时间超出了指定的时间为止。

【关于IOCP的文摘】

Q：服务程序中并发线程数如何确定？

A：An ideal model would strike a balance between the two extremes. There should always be enough runnable threads to fully utilize the available CPUs, but there should never be so many threads that the overhead becomes too large. In fact, the ideal number of runnable threads is not related to the number of clients at all, but to the number of CPUs in the server.