第10章 同步设备I/O与异步设备I/O

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

（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。