1. 线程调度：
   1. 线程优先级

THREAD\_PRIORITY\_IDLE 懒惰优先级

THREAD\_PRIORITY\_LOWEST 低优先级

THREAD\_PRIORITY\_BELOW\_NORMAL 正常低优先级

THREAD\_PRIORITY\_NORMAL 正常优先级

THREAD\_PRIORITY\_ABOVE\_NORMAL 正常高优先级

THREAD\_PRIORITY\_HIGHEST 高优先级

THREAD\_PRIORITY\_TIME\_CRITICAL 实时优先级

* 1. 进程优先级类

IDLE\_PRIORITY\_CLASS 懒惰优先级类

BELOW\_NORMAL\_PRIORITY\_CLASS 正常低优先级类

NORMAL\_PRIORITY\_CLASS 正常优先级类

ABOVE\_NORMAL\_PRIORITY\_CLASS 正常高优先级类

HIGH\_PRIORITY\_CLASS 高优先级了类

REALTIME\_PRIORITY\_CLASS 实时优先级类

* 1. 线程调度优先级

线程调度优先级分为0-31级，0级最底，31级最高。操作系统内核中的线程调度程序使用线程调度优先级来决定那个线程运行，那个线程等待。（其中运行在0优先级的只有一个线程，这个线程用来将空闲内存清零）。

如果有高优先级的线程需要运行，那么低优先级的线程永远得不到运行的机会。同优先级的线程会轮流被执行。

线程的调度优先级如何确定：

|  |  |  |
| --- | --- | --- |
| **Process priority class** | **Thread priority level** | **Base priority** |
| IDLE\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 1 |
| THREAD\_PRIORITY\_LOWEST | 2 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 3 |
| THREAD\_PRIORITY\_NORMAL | 4 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 5 |
| THREAD\_PRIORITY\_HIGHEST | 6 |
| THREAD\_PRIORITY\_TIME\_CRITICAL | 15 |
| BELOW\_NORMAL\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 1 |
| THREAD\_PRIORITY\_LOWEST | 4 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 5 |
| THREAD\_PRIORITY\_NORMAL | 6 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 7 |
| THREAD\_PRIORITY\_HIGHEST | 8 |
| *THREAD\_PRIORITY\_TIME\_CRITICAL* | *15* |
| * NORMAL\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 1 |
| THREAD\_PRIORITY\_LOWEST | 6 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 7 |
| THREAD\_PRIORITY\_NORMAL | 8 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 9 |
| THREAD\_PRIORITY\_HIGHEST | 10 |
| THREAD\_PRIORITY\_TIME\_CRITICAL | 15 |
| ABOVE\_NORMAL\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 1 |
| THREAD\_PRIORITY\_LOWEST | 8 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 9 |
| THREAD\_PRIORITY\_NORMAL | 10 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 11 |
| THREAD\_PRIORITY\_HIGHEST | 12 |
| THREAD\_PRIORITY\_TIME\_CRITICAL | 15 |
| HIGH\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 1 |
| THREAD\_PRIORITY\_LOWEST | 11 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 12 |
| THREAD\_PRIORITY\_NORMAL | 13 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 14 |
| THREAD\_PRIORITY\_HIGHEST | 15 |
| THREAD\_PRIORITY\_TIME\_CRITICAL | 15 |
| REALTIME\_PRIORITY\_CLASS | THREAD\_PRIORITY\_IDLE | 16 |
| THREAD\_PRIORITY\_LOWEST | 22 |
| THREAD\_PRIORITY\_BELOW\_NORMAL | 23 |
| THREAD\_PRIORITY\_NORMAL | 24 |
| THREAD\_PRIORITY\_ABOVE\_NORMAL | 25 |
| THREAD\_PRIORITY\_HIGHEST | 26 |
| THREAD\_PRIORITY\_TIME\_CRITICAL | 31 |

注意区分中断优先级和线程优先级。

线程的调度程序运行在DPC中断优先级上，中断优先级也是0-31。中断优先级不是线程优先级。

中断优先级列表：

31：High

30：Power fail

29：Interprocessor interrupt

28：Clock

27：Profile/Synch

26：Device n

5： Corrected Machine Check Interrupt

4：

3： Device1

2： DPC/dispatch

1： APC

0: Passive/Low

其中 3-31都是硬件中断

2-1 是软件中断

0： 通常线程都在这个中断等级上执行。

CreateThread Openprocess

**[SetPriorityClass](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=02686637-427A-4CF1-A4E5-60C707AF3084&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)**

[**GetPriorityClass**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=2A16B18F-8EFA-43F0-9F7D-D38CC8A153D3&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**SetThreadPriority**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=E3992E19-B546-4B0B-AA6A-DD9A7E330BF3&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**GetThreadPriority**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=E3992E19-B546-4B0B-AA6A-DD9A7E330BF3&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**GetProcessAffinityMask**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=F50CA86E-FA81-4ED9-AE6C-63A4E7F2A53F&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**SetProcessAffinityMask**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=F50CA86E-FA81-4ED9-AE6C-63A4E7F2A53F&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**SetThreadAffinityMask**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=3390930D-026F-4F86-97BC-1DA34BB384BA&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

[**GetThreadAffinityMask**](http://127.0.0.1:47873/help/1-7668/ms.help?method=page&id=3390930D-026F-4F86-97BC-1DA34BB384BA&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)

1. **同步原理**

**最基本的原理：不停查看某个状态，如果满足状态，接着干活，否则接着查看！**

**计算机实现同步的方式应该注意哪些问题。**

**多CPU之间的互锁（interlocked系列函数）**

**使用最基本同步原理产生的问题。**

**等待线程在不停的做无用功，也就是我们通常所说的磨洋工，只干活，不出活。**

**如何解决这些问题。**

**叫领导来！！！**