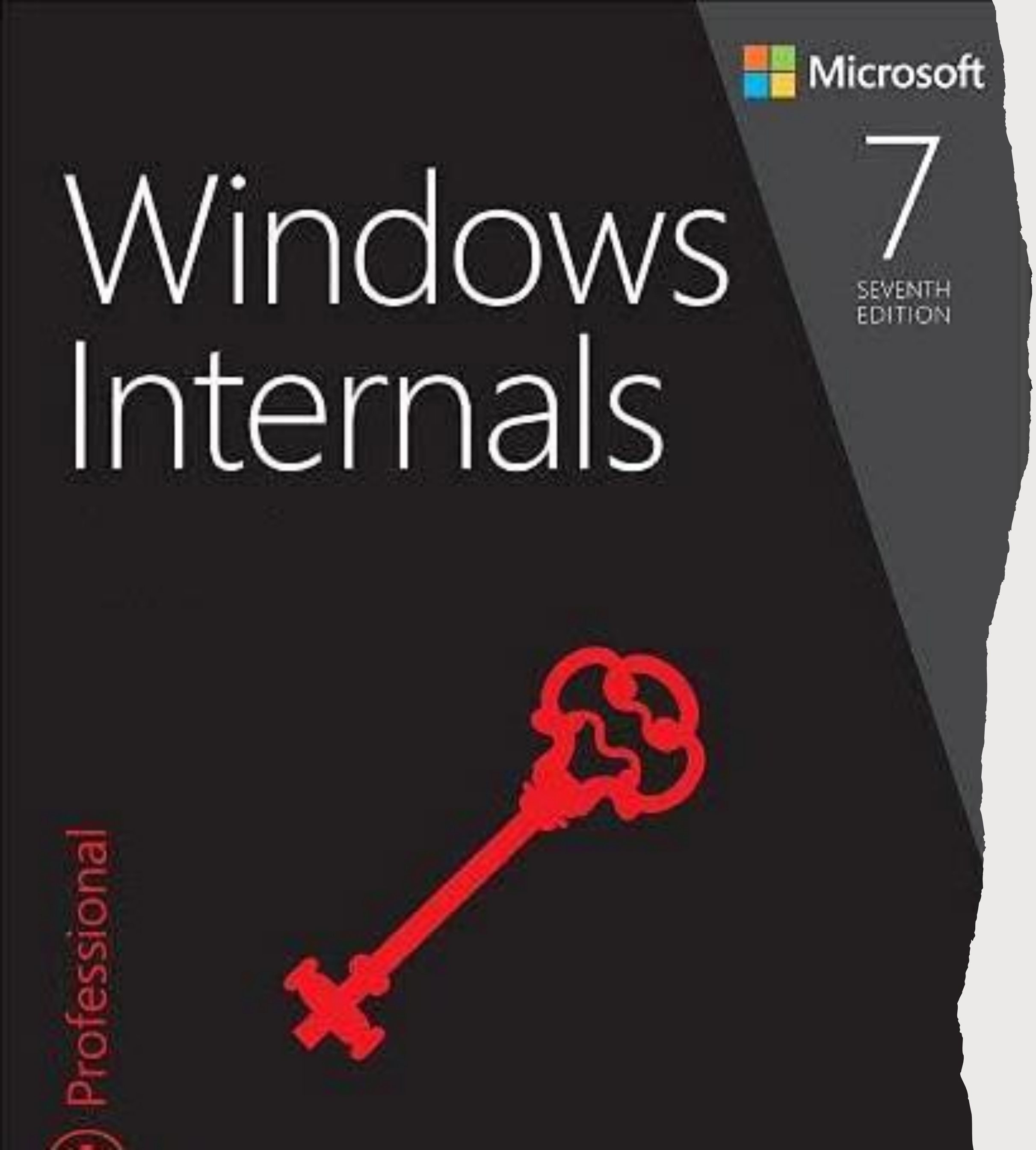


01)Threading

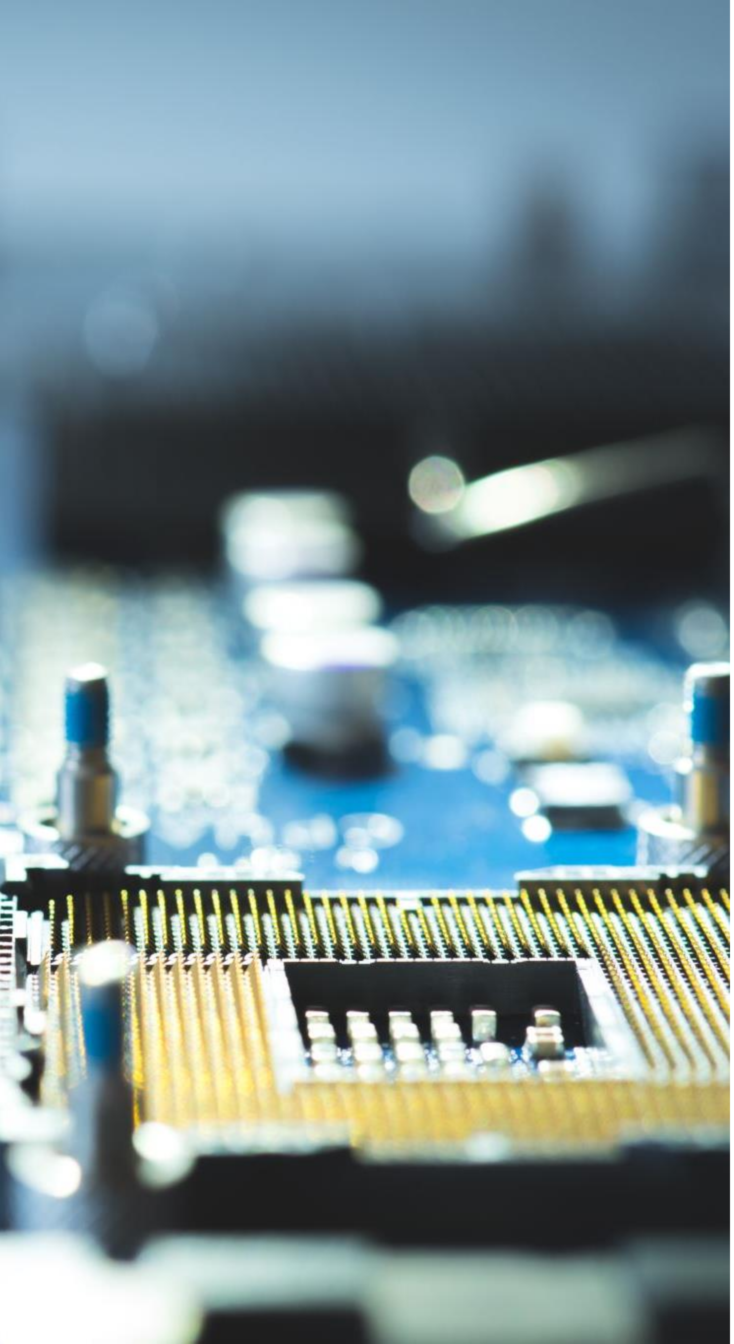
Thread Scheduling

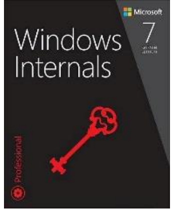
Scheduling Basics

Priority

Affinity

Adjustments



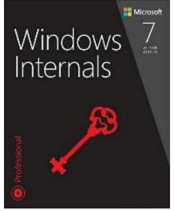


Topics To Cover

- Thread Priorities
- Thread States
- Priority Boosts
- Affinity
- CPU Sets

akazemi67

Thread Priorities



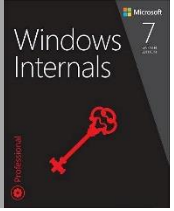
Ranging from 0 to 31, where 31 is the highest

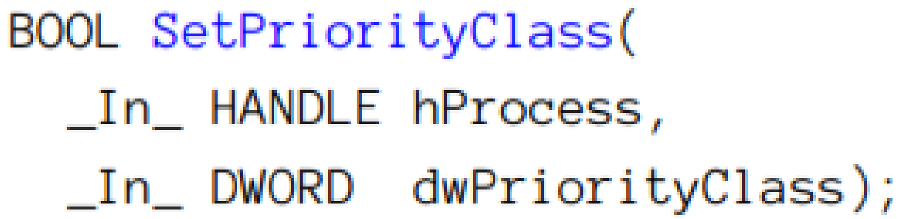
Priority 0 is reserved for **zero page** thread (part of the memory manager in the kernel)

Thread's priority is a combination of a process' Priority Class (called Base Priority) and an offset around that base

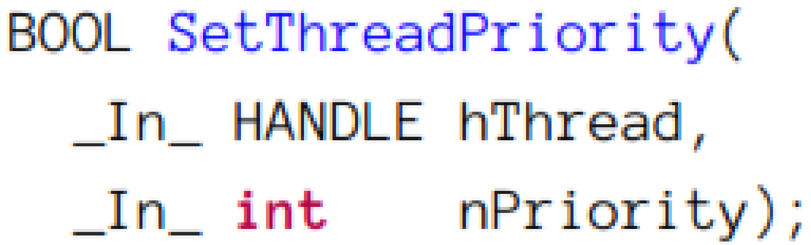
Each priority class is associated with a **priority value**

APIs for Changing Priority





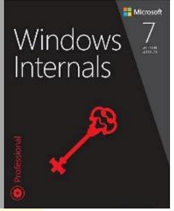
```
BOOL SetPriorityClass(
   _In_ HANDLE hProcess,
   _In_ DWORD  dwPriorityClass);
```

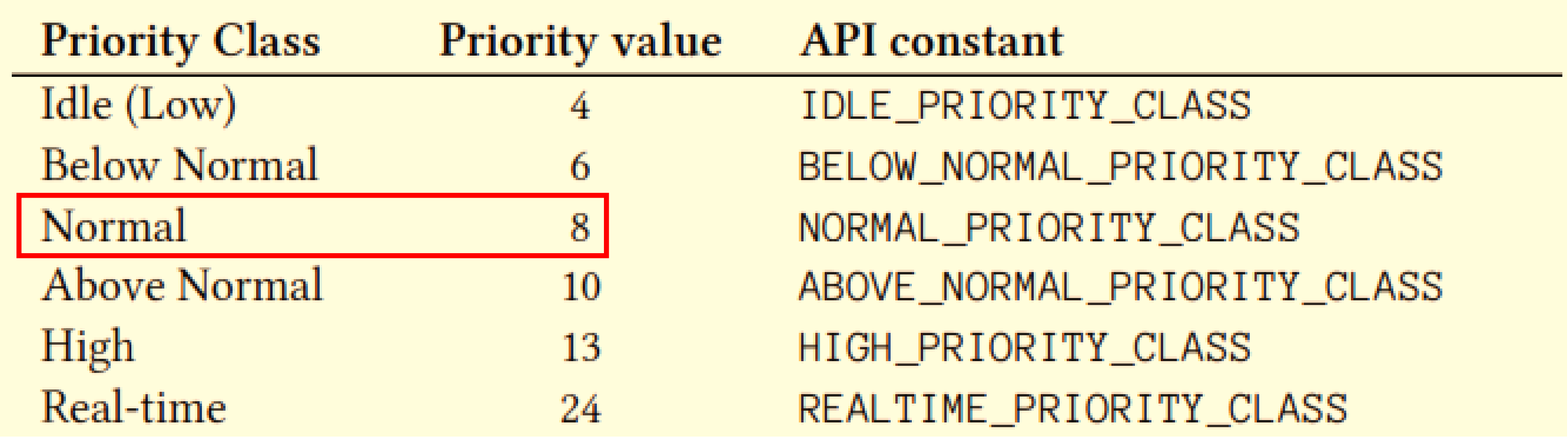


```
BOOL SetThreadPriority(
   _In_ HANDLE hThread,
   _In_ int    nPriority);
```

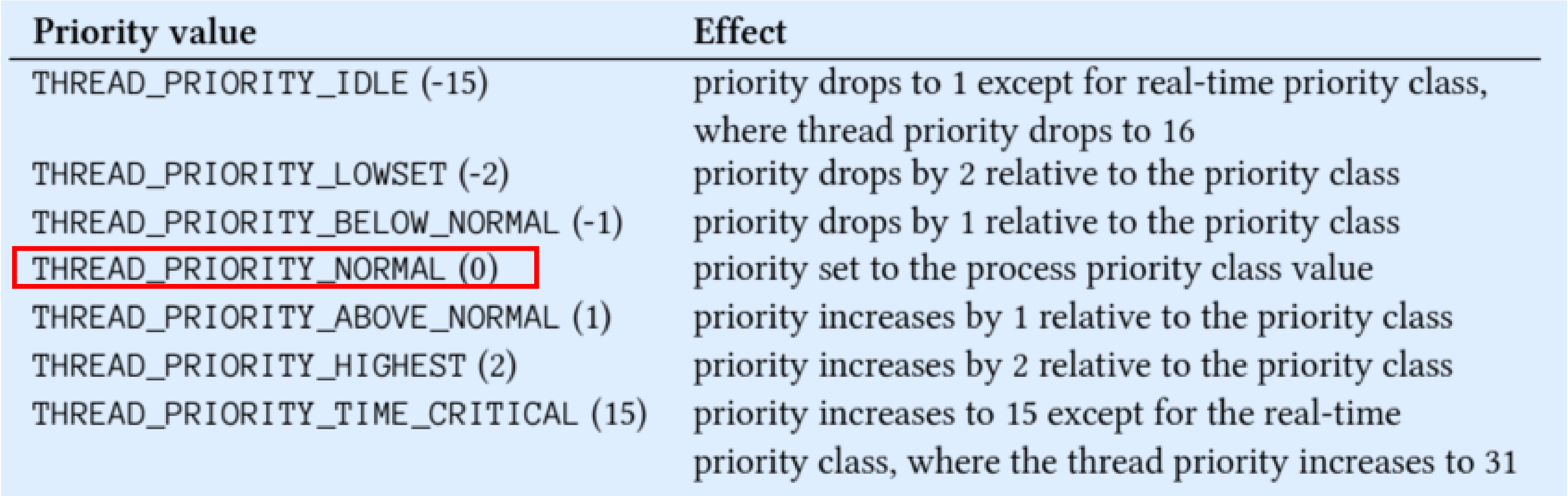
Note: Kernel can set a thread’s priority to any number, regardless of its process priority class.

Priority Classes





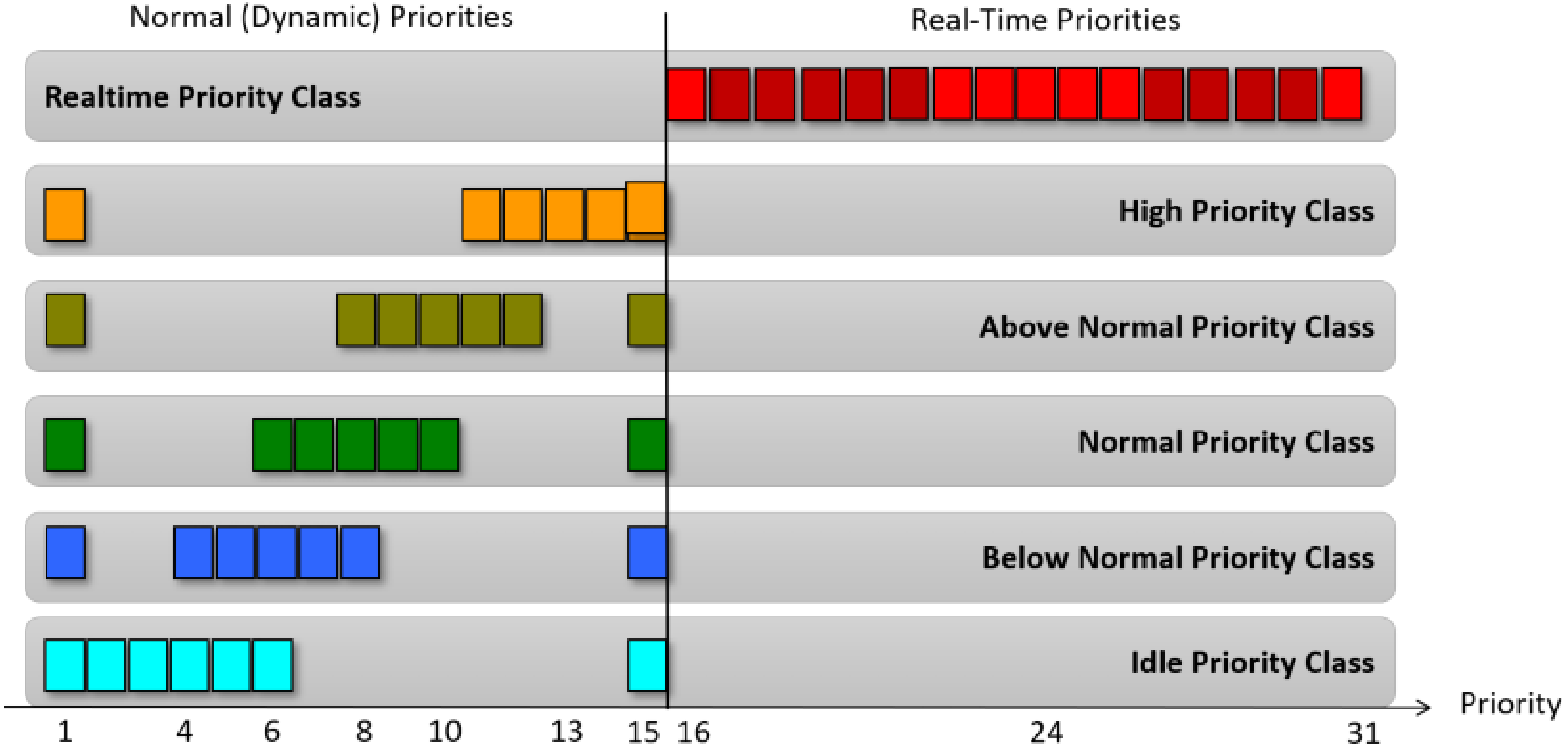
Priority Class	Priority value	API constant
Idle (Low)	4	IDLE_PRIORITY_CLASS
Below Normal	6	BELOW_NORMAL_PRIORITY_CLASS
Normal	8	NORMAL_PRIORITY_CLASS
Above Normal	10	ABOVE_NORMAL_PRIORITY_CLASS
High	13	HIGH_PRIORITY_CLASS
Real-time	24	REALTIME_PRIORITY_CLASS



Priority value	Effect
THREAD_PRIORITY_IDLE (-15)	priority drops to 1 except for real-time priority class, where thread priority drops to 16
THREAD_PRIORITY_LOWSET (-2)	priority drops by 2 relative to the priority class
THREAD_PRIORITY_BELOW_NORMAL (-1)	priority drops by 1 relative to the priority class
THREAD_PRIORITY_NORMAL (0)	priority set to the process priority class value
THREAD_PRIORITY_ABOVE_NORMAL (1)	priority increases by 1 relative to the priority class
THREAD_PRIORITY_HIGHEST (2)	priority increases by 2 relative to the priority class
THREAD_PRIORITY_TIME_CRITICAL (15)	priority increases to 15 except for the real-time priority class, where the thread priority increases to 31

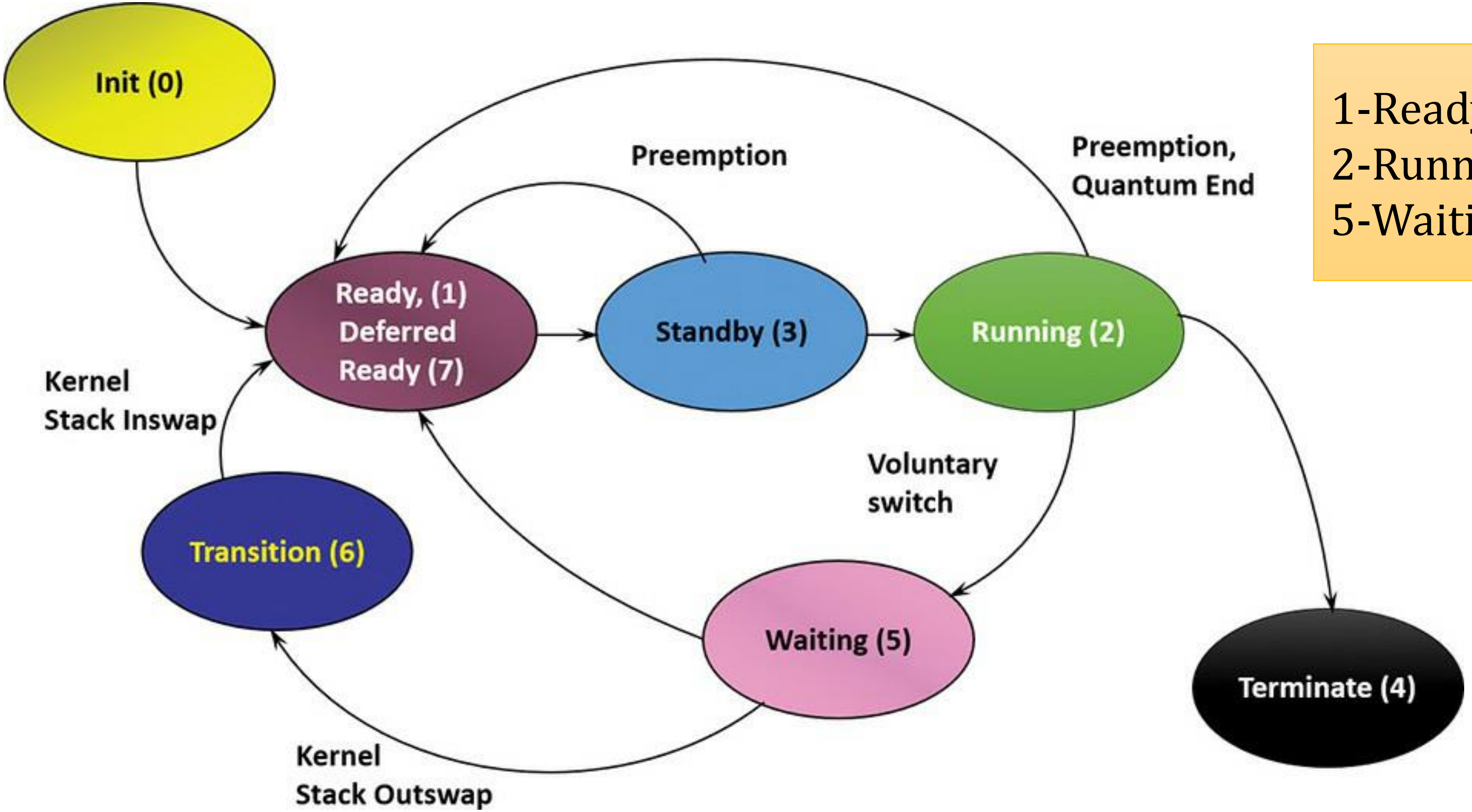
akazemi67

Priority Values



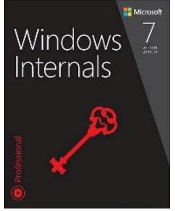
akazemi67

Thread States



1-Ready
2-Running
5-Waiting

akazemi67



Priority Boosts

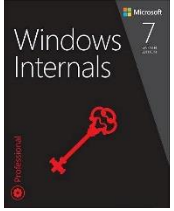
The Windows scheduler periodically adjusts the current (dynamic) priority of threads to:

- Decrease various latencies
- Increase responsiveness

In other words, it applies these boosts:

- To prevent inversion
- To prevent starvation

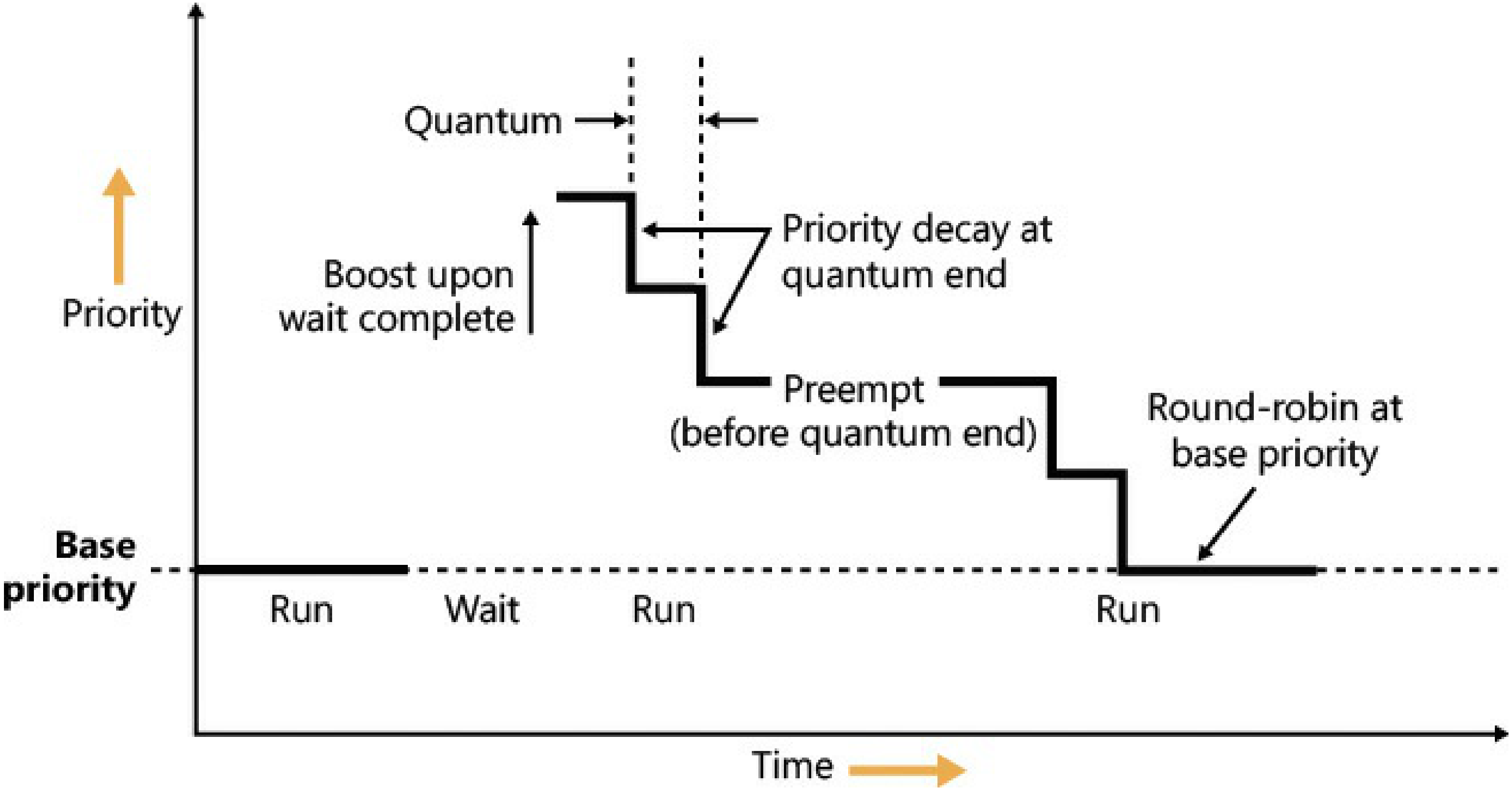
Boost Scenarios



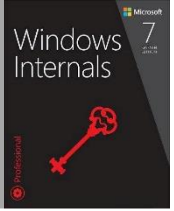
- User interface (UI) input (latency reduction/responsiveness)
- I/O completion (latency reduction)
- Thread waiting on an executive resource for too long (starvation avoidance)
- A ready thread hasn’t been running for some time (starvation and priority inversion avoidance)

Note: Windows never boosts the priority of threads in the real-time range (16 through 31)

Priority Boosting and Decay

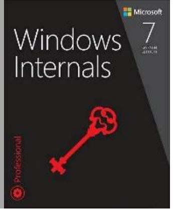

Quantum
Priority
Boost upon wait complete
Priority decay at quantum end
Preempt (before quantum end)
Round-robin at base priority
Base priority
Run
Wait
Run
Run
Time

Affinity



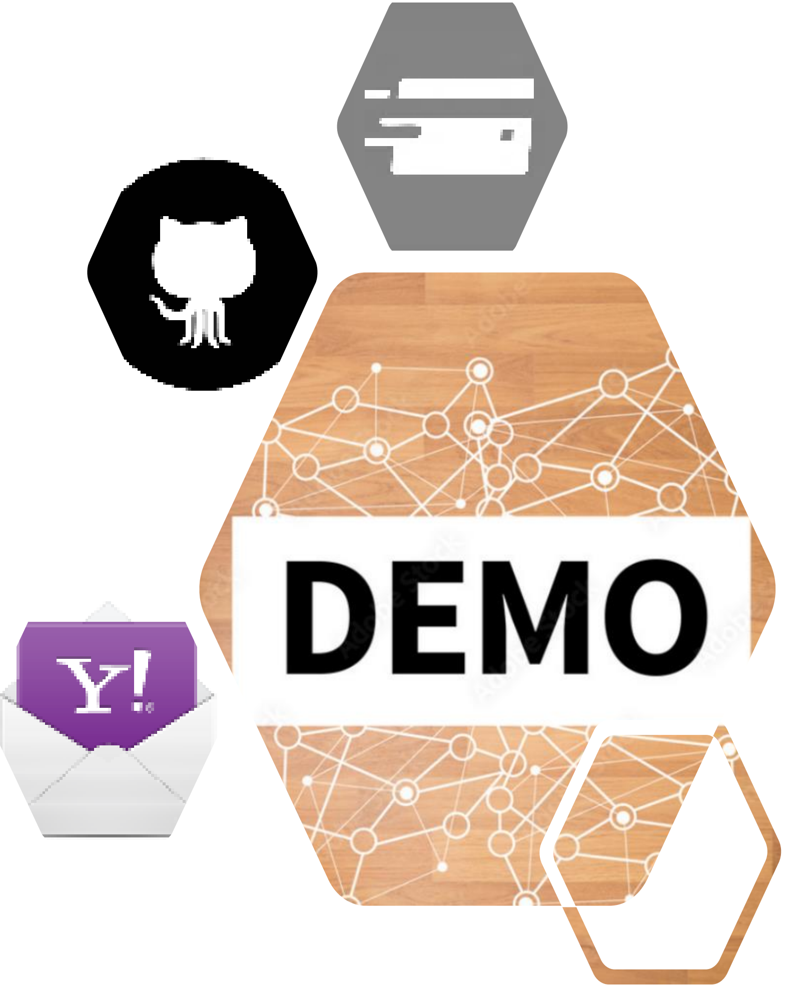
- The *Ideal Processor* is an attribute of a thread, also sometimes referred to as “Soft Affinity”
- The affinity of a thread is the processors it’s allowed to run on (Hard Affinity)
- Hard affinity works on two levels: process and thread
- A thread cannot “escape” the affinity set by its process.

CPU Sets



- There are some scenarios where it’s beneficial to have a thread (or threads) use processors that other threads in the process are forbidden to use
- Windows 10 and Server 2016 added this capability
- If CPU sets contradicts hard affinity, CPU sets are ignored.

DEMO



Thread Scheduling

akazemi67