# System Idle Process

In Windows NT operating systems, the **System Idle Process** contains one or more kernel threads which run when no other runnable thread can be scheduled on a CPU. For example, there may be no runnable thread in the system, or all runnable threads are already running on a different CPU. In a multiprocessor system, there is one idle thread associated with each CPU.

The primary purpose of the idle process and its threads is to eliminate what would otherwise be a special case in the scheduler. Without the idle threads, there could be cases when no threads were runnable, or "Ready" in terms of Windows scheduling states. Since the idle threads are always in a Ready state (if not already Running), this can never happen. Thus whenever the scheduler is called due to the current thread leaving the CPU, it can always find another thread to run on that CPU, even if it is only the CPU's idle thread.

The scheduler will never select an idle thread for execution on a given CPU if there are any other threads eligible to run on that CPU. The CPU time attributed to the idle process is therefore indicative of the amount of CPU time that is *not* needed or wanted by any other processes in the system. The scheduler treats the idle threads as special cases in terms of thread scheduling priority. All other threads have a priority in the range 0 through 31, inclusive, with higher priorities always preempting lower. The idle threads do have a priority member in their thread objects but this data is not used to determine when to run them. Instead the scheduler selects the idle thread for a CPU whenever there are no threads in the normal priority range that can run on the CPU. It is as if the idle threads each have a priority of −1. It is not possible to create additional threads that are scheduling peers of the idle threads, even from kernel mode.

Because of the idle process's function, its CPU time measurement (easily available through Windows Task Manager) may make it appear to users that the idle process is monopolizing the CPU. However, it is actually a simple measure of the system CPU utilization. The System Idle Process and its threads eliminate the possibility of the scheduler having to deal with the exception to a rule, thanks to their special scheduling priority. If no ordinary thread is able to run on a free CPU, then and only then does the scheduler select that CPU's System Idle Process thread for execution. The idle process, in other words, is merely acting as a sort of placeholder during "free time" (therefore, whenever the idle process appears to be consuming most of the CPU, it is proof that no other process wants that CPU time).

In Windows 2000 and later the threads in the System Idle Process are also used to implement CPU power saving. The exact power saving scheme depends on the operating system version and on the hardware and firmware capabilities of the system in question. For instance, on x86 processors under Windows 2000, the idle thread will run a loop of halt instructions, which causes the CPU to turn off many internal components until an interrupt request arrives. Later versions of Windows implement more complex CPU power saving methods. On these systems the idle thread will call routines in the Hardware Abstraction Layer to reduce CPU clock speed or to implement other power-saving mechanisms.

There are more detailed sources of such information available through Windows' performance monitoring system (accessible with the perfmon program), which includes more finely grained categorization of CPU usage. A limited subset of the CPU time categorization is also accessible through the Task Manager, which can display CPU usage by CPU, and categorized by time spent in user vs. kernel code.