# CS490 Windows Internals

# Assignment 2

# Due: Friday, Oct. 11, 2013

1. Give a short explanation of the following terms and its usage in context of the Windows operating system to your instructor:

* Handle
* Process
* Thread

Sol.

* Handle
  + In the Windows operating system, an object is a single, run-time instance of a statically defined object type. An object type comprises a system-defined data type, functions that operate on instances of the data type, and a set of object attributes. User programs may acquire access to operating system objects via *handles*.
* Process
  + Represents an instance of a running program
  + You create a process to run a program
  + Starting an application creates a process
  + Process defined by:
    - Address space
    - Resources (e.g. open handles)
    - Security profile (token)
* Thread
  + An execution context within a process
  + Unit of scheduling (threads run, processes don’t run)
  + All threads in a process share the same per-process address space

1. Describe the five-state model for process/thread scheduling. What are the benefits/rationales for multithreaded programming? How many threads does a Windows process initially contain?

Sol.

The five-state model for process/thread scheduling can be described as follows: Newly created threads start there live in “init” state. After appropriate data structures are initialized (thread control block - TCB), the thread’s state is changed to “ready”. From “ready” state, a thread may get selected for execution and its state is changed by the scheduler to “running”. For a given system, there is exactly one thread in “running” state per processing unit.

When the “running” thread issues an I/O request, it might block and change its state into “waiting”. Alternatively, a “running” thread’s time slice (quantum) may expire, in which case the thread’s state is changed back to “ready”. A “waiting” thread can move into “ready” state once the reason for its blocking disappears (I.e.; the I/O operation completes). Finally, a “running” thread might execute its last instruction - in which case its state changes to “terminated”.

The benefits of multithreaded programming are:

* Higher Responsiveness by using dedicated threads for handling user events
* Simpler Resource Sharing since all threads in a process share same address space
* Utilization of Multiprocessor Architectures
* Multiple threads may run in parallel

Each Windows process initially contains one thread.

1. What are DPCs and APCs? Describe one typical scenario for each of these Windows mechanisms. Use diagrams if necessary.

Sol.

DPC: A routine that performs most of the work involved in handling a device interrupt after the interrupt service routine (ISR) executes. The DPC routine executes at an interrupt request level (IRQL) that is lower than that of the ISR to avoid blocking other interrupts unnecessarily. A DPC routine initiates I/O completion and starts the next queued I/O operation on a device.

APC: A function that provides a way for user programs and system code to execute code in the context of a particular user thread (and hence a particular process address space). An APC can be either kernel mode or user mode. (Kernel-mode APCs don’t require “permission” from a target thread to run in that thread’s context, as user-mode APCs do.)