The purpose of this paper is to provide a concise summary of the Windows Executive. The term, Windows Executive, is seldom used among most computer users, and therefore a brief definition of this term, and a few related terms is necessary before proceeding.

Microsoft defines the Windows Executive as “kernel mode components that provide a variety of services to device drivers, including object management, memory management, process and thread management, input/output management, and configuration management.” Kernel mode refers to one of two modes in which a processor runs in a Windows computing environment, user mode being the other. Because the explanations of both kernel and user modes easily constitute individual presentations, only a brief explanation is given.

User mode is self-explanatory; the user’s applications and interface, plus some (but not all) drivers, and all routines related to the current state of the computer reside within the user mode. One may think of it as the “workspace,” and it primarily consists of the “what” regarding a user’s activities with a Windows based PC.

In contrast, kernel mode is the “how” regarding the user’s activities, and includes the hardware abstraction layer, file system and other kernel mode drivers, and Exported Driver Support Routines. According to Microsoft Developer Hany Barakat, in kernel mode the operating “code has direct access to the hardware.” Within kernel mode, the file-system drivers, other kernel mode drivers, and the Exported Driver Support Routines consist of the Windows Executive.

There is some confusion regarding which components consist of the Windows Executive, however. While the above definition of the Windows Executive is consistent with Barakat’s description, documentation from Microsoft states that the Windows Executive is apart from the drivers, providing services, but separate. Because the Windows Executive and the drivers together comprise one subsystem within the Operating System, for the purposes of this paper both the drivers and the Windows Executive are one. However, this example is one of several which complicate the subject of the Windows Executive, rendering research of the topic difficult.

Within the Windows Executive, several subsystems are present: **The Object Manager, I/O Manager, Security Reference Monitor, IPC Manager, Virtual Memory Manager, Process Manager, PnP Manager,** and the **Power Manager.** At least one resource lists the **Window Manager**, and the **Graphics Device Interface,** but as these managers are related to providing output on a desktop monitor, they are part of the I/O Manager, and do not require a separate section within this paper. Each subsystem has unique tasks and attributes, examined in the following sections.

The **Object Manager** creates objects, consisting of a standard header and possessing attributes specific to each object. Note “object” in Windows computing refers to a data structure, and may represent any number of items, such as files, threads, or graphical images. (Microsoft). In addition to creating the object, the Object Manager is responsible for verifying that processes have the right to use an object, it maintains resources quotas, creates object handles, duplicate handles, and closes the handles upon conclusion, (Microsoft).

The **I/O (Input/Output) Manager** “manages the communication between applications and the interfaces provided by device drivers,” (Microsoft), in concise terms, the I/O Manager is responsible for handing the communication between the Operating System and all Input or Output devices attached to the PC, such as keyboards, printers, and speakers. Within the I/O Manger is the **Plug and Play (PNP) Manager**, mentioned above, which “is a combination of hardware technology and software techniques that enable a PC to recognize when a device is added to the system, (Microsoft). Simply stated, the PnP Manager is responsible for controlling, or managing, the addition of I/O devices to the system without requiring interaction from the user, or if setup requires interaction, with minimal input on the part of the user. A second component of the I/O Manager is the **Power Manager**, which manages the change in power state for all devices supporting such change, such as external monitors, (Microsoft).

The **Security Reference Monitor** manages the security between objects and action requests. Providing routines to drivers, the Security Reference Monitor allows system devices to access the Windows Access Control List, which determines “which objects have what security,” (Microsoft).

The **IPC (Interprocess Communications) Manager** is a mechanism which facilitates communication and data sharing between applications, such as Microsoft Office applications, typically in a client (requesting) or server (provider) role, (Microsoft).

The **Virtual Memory Manager** controls how applications manage virtual memory in Windows, allowing applications to execute, and manage their own address space, without the need to “consider the impact on other processes in the system,” (Microsoft). Because of the Virtual Memory Manager, applications are able reserve, commit, free, change, lock and query the virtual memory they require.

The **Process Manager** is the manager for all thread and process creation or termination. Note that within the scope of this paper, threads and processes refer to user actioned and system applications that run either manually or automatically.

Works Cited

Architecture of Windows NT, McGill School of Computer Science, www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/a/Architecture\_of\_Windows\_NT.htm.

Barakat, Hany. “Deeper into Windows Architecture.” Hany Barakat's Technical Blog, Microsoft Corporation, 25 Feb. 2007, blogs.msdn.microsoft.com/hanybarakat/2007/02/25/deeper-into-windows-architecture/.

“Handles and Objects.” Handles and Objects (Windows), Microsoft Corporation, msdn.microsoft.com/en-us/library/windows/desktop/ms724457(v=vs.85).aspx.

“Interprocess Communications.” Interprocess Communications (Windows), Microsoft Corporation, msdn.microsoft.com/en-us/library/windows/desktop/aa365574(v=vs.85).aspx.

“Managing Virtual Memory.” Managing Virtual Memory, Microsoft Corporation, msdn.microsoft.com/en-us/library/ms810627.aspx.

“Object Manager.” Object Manager (Windows), Microsoft Corporation, msdn.microsoft.com/en-us/library/windows/desktop/ms724826(v=vs.85).aspx.

“User Mode and Kernel Mode.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode.

“Windows Kernel-Mode Executive Support Library.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-executive-support-library.

“Windows Kernel-Mode I/O Manager.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-i-o-manager.

“Windows Kernel-Mode Plug and Play Manager.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-plug-and-play-manager.

“Windows Kernel-Mode Power Manager.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-power-manager.

“Windows Kernel-Mode Security Reference Monitor.” Microsoft Docs, Microsoft Corporation, docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-security-reference-monitor.