Evolution of the Microsoft Windows Operating System and Graphical User Interface

Joseph E. Gann

XMIT 604: Computing & Software Technology

Dr. Henry Tsai

Introduction – What is An Operating System?

This paper will discuss the evolution of the Microsoft Windows Operating System, its user interface, and its interaction with a computer's architecture. Tanenbaum, in his 2006 book *Structured Computer Organization* defines an operating system as:

> … a program that, from the programmer's point of view, adds a variety of new instructions and features, above and beyond what the Instruction Set Architecture (ISA) level provides. It creates an Operating System Machine abstract where the programmer can call ISA instructions and operating system-level system calls. It is an "extended machine" which: hides the messy details which must be performed, presents the user with an easier to use virtual machine, is a resource manager where each program gets allocated time and space (pp. 426-27).

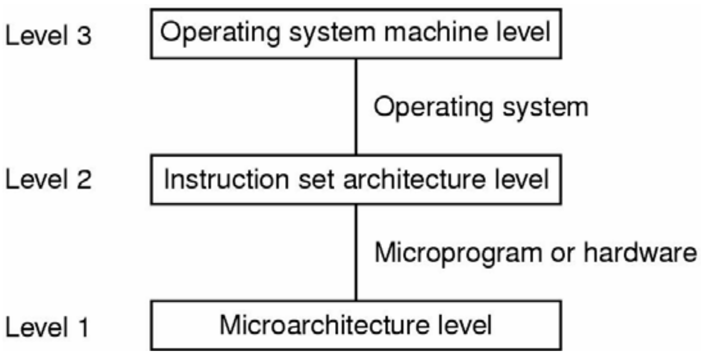


Figure 1: Operating system in relationship to Tanenbaum's computer (Tanenbaum, p. 428)

The operating system level itself contains layers. The lowest level of any operating system is its kernel. This is the first layer of software loaded into memory when a system boots or starts up. The kernel provides access to various common core services to all other system and application programs. These services include, but are not limited to: task scheduling, memory management, disk access, and access to hardware devices (http://en.wikipedia.org/wiki/Operating_System, 2006). As well as the kernel, an operating system is often distributed with system software that manages a graphical user interface (although Windows, which will be discussed, and Macintosh have integrated these programs into the operating system); as well as, utility programs for tasks such as managing files and configuring the operating system.

Thus, at the simplest level, an OS performs two tasks: manages the hardware and software resources of the system, and it provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware. The first task, managing the hardware and software resources, is very important, as various programs and input methods compete for the attention of the central processing unit (CPU) and demand memory, storage and input/output (I/O) bandwidth for their own purposes. In this capacity, the operating system plays the role of the good parent, making sure that each application gets the necessary resources while playing nicely with all the other applications; as well as, husbanding the limited capacity of the system to the greatest good of all the users and applications (Coustan & Franklin, 2006). The second task, providing a consistent application interface, is especially important if there is to be more than one of a particular type of computer using the operating system, or if the hardware making up the computer is ever open to change. A consistent application program interface (API) allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines (Coustan & Franklin, 2006).

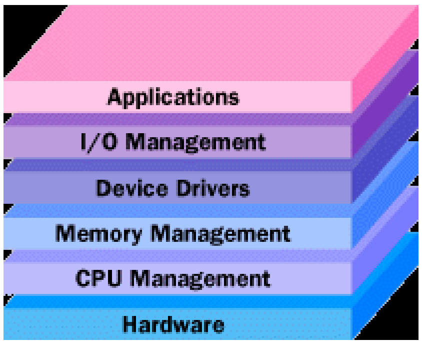


Figure 2: Levels of an Operating System (Coustan & Franklin, 2006)

The first computers did not have operating systems. However, software tools for managing the system and simplifying the use of hardware appeared very quickly afterwards and gradually expanded in scope. By the early 1960s, commercial computer vendors were supplying quite extensive tools for streamlining the development, scheduling, and execution of jobs on batch processing systems. Through the 1960s, several major concepts were developed, driving the development of operating systems (http://en.wikipedia.org/wiki/Operating_System, 2006).

The first microcomputers did not have the capacity or need for the elaborate operating systems that had been developed for mainframes and minis; minimalist operating systems were developed, beginning with MS-DOS and then the first variation of windows.

WIMP and the beginning of GUI

One of the most impressive developments in software during the past two decades has been the triumph of the Windows, Icons, Menus, Pointing interface (WIMP). First displayed by Doug Englebart of Stamford Research Institute, it was purchased by Xerox and eventually one Bill Gates – who transformed this concept into the Window OS (Brooks Jr., 1995, p. 260).

The WIMP is a superb example of a user interface that has conceptual integrity, achieved by the adoption of a familiar mental model, the desktop metaphor. This metaphor was built upon sizable and movable windows, drag-and-drop, point at and selecting icons, nested folders, cut-copy-paste, and the recycle bin and directly linked to common tasks performed at a real desk (Brooks Jr., 1995, p. 261).

The Windows Operating System – Prior to XP

In the beginning, there was Windows 1.0, and it was good. This next portion of this paper will discuss the evolution of Windows from 1.0 to 2000/ME and each version's relationship with the machine's architecture.

Windows 1.0 – October 1985

Windows 1.0 offered limited multitasking of existing MS-DOS programs and concentrated on creating an interaction paradigm, an execution model and a stable API for native programs for the future. Windows 1.0 programs could call MS-DOS functions, and GUI programs were run from .exe files just like MS-DOS programs. However, Windows .exe files had their own "new executable" (NE) file format, which only Windows could process and which for example allowed demand-loading of code and data. Applications were supposed to handle memory only through Windows' own memory management system, which implemented a software-based virtual memory scheme allowing for applications larger than available RAM. Windows 1.0 included original device drivers for video cards, mice, keyboards, printers and serial communications. Applications were supposed to only invoke APIs built upon these drivers. Given that contemporary graphics support in MS-DOS was extremely limited and given the limited usefulness of the other services, MS-DOS applications had to go to the bare hardware (or sometimes just to the BIOS) to get work done.

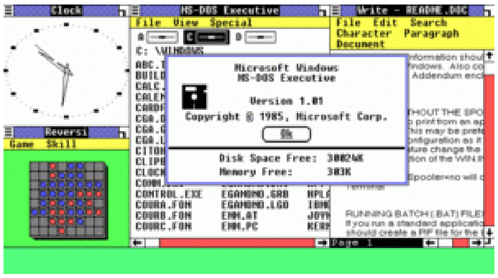
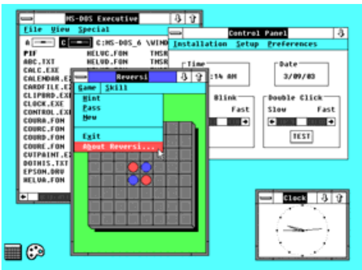


Figure 3: Typical Windows 1.0 desktop (http://en.wikipedia.org/wiki/Windows_1.0, 2006).

Windows 1.0 executables, while having the same .exe extension and initial file header as MS-DOS programs, did not yet contain the so-called MS-DOS stub which prints the "This program requires Microsoft Windows" message and exits when the program is run outside of Windows. Instead, the file header was formatted in such a way as to make DOS reject the executable with a "program too large to fit in memory" error message (http://en.wikipedia.org/wiki/Windows_1.0, 2006).

Windows 2.0 – November 1987

Figure 4: Windows 2 Desktop (http://en.wikipedia.org/wiki/Windows_2.0)



There were not too many major changes to the Windows 2.0 architecture as most functions were still tied to MS-DOS and applications programmers and users preferred to use DOS-based programs.

Windows 2.1x (Windows 286 and 386) – May 1998

Windows/286 took advantage of the HMA (High memory Area) to increase the memory available to Windows programs. It introduced the himem.sys DOS driver for this purpose. It also included support for several

EMS (extended memory specification) boards for this same purpose (this support was not related to the 80286 processor per-se). The segmented nature of Windows programs was quite suited to the usage of EMS, as portions of code and data could be made visible in the first megabyte of memory accessible to real-mode programs only when the program using them was given control. Microsoft encouraged users to configure their computers with only 256KB of main memory, leaving the address space from 256-640KB available for dynamic mapping of EMS memory (http://en.wikipedia.org/wiki/Windows_2.0, 2006).

Windows/386 was much more innovative. It introduced a protected mode kernel, above which the GUI and applications were running as a virtual 8086 mode task. It allowed several MS-DOS programs to run in parallel in virtual machines, rather than always suspending background applications. Windows applications could already run in parallel through cooperative multitasking. Each DOS application could use as much low memory as was available before Windows was started, minus a few kilobytes of overhead. Windows also provided EMS emulation, using the memory management features of the processor to make RAM beyond 640K behave like the banked memory previously only supplied by add-in cards and used by popular DOS applications. There was no disk-based virtual memory, so multiple DOS programs had to fit inside the available physical memory; Microsoft suggested buying additional memory (and cards) if necessary (http://en.wikipedia.org/wiki/Windows_2.0, 2006).

Windows 3.0 – May 1990

Windows 3.0 included a significantly revamped user interface as well as technical improvements to make better use of the memory management capabilities of Intel's 80286 and 80386 processors. Text-mode programs written for MS-DOS could be run within a window, making the system usable as a crude multitasking base for legacy programs. However, this was of limited use for the home market, where most games and entertainment programs continued to require raw DOS access. The MS-DOS Executive file manager/program launcher was replaced with an icon-based Program Manager and a list-based File Manager, thereby simplifying the launching of applications. The MS-DOS Executive was still included as an alternative user interface program. The Control Panel, previously available as a standard-looking applet centralizing the various system settings (http://en.wikipedia.org/wiki/Windows_3.0, 2006).

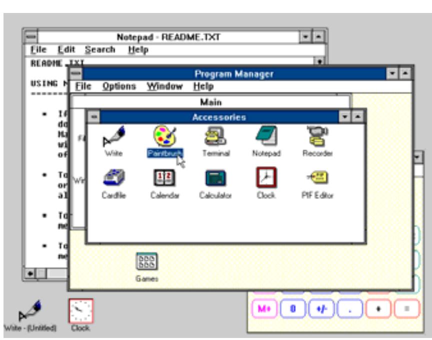
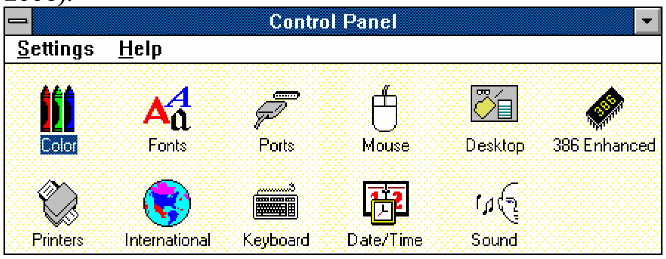


Figure 5: Windows 3.0 Desktop and applications (http://en.wikipedia.org/wiki/Windows_3.0, 2006).

Figure 6 (below): Windows 3.0 Control panel – new in this version (http://en.wikipedia.org/wiki/Windows_3.0, 2006).



Windows 3.0 was the only version of Windows that could be run in three different memory modes:

- Real mode, intended for older computers with a CPU below Intel 80286 Real mode set the CPU to run in real mode, as though it were an Intel 80186, including the limitation that it could only address 1 MB of RAM. The expanded memory scheme was used to utilize any memory the computer had beyond 1 MB. This slowed down the computer significantly, and was used only by users of legacy applications that would crash in protected mode. Windows 3.0 was the last version of Windows that could run in real mode.
- Standard mode, intended for computers with an 80286 processor, and corresponding to its protected mode. Standard mode on a 286 or later switches the CPU to protected mode, and therefore let the CPU directly access up to 16 MB of RAM at once, enabled virtual memory, and used memory protection to make Windows more stable in the event of an application fault. Support for standard mode was dropped in Windows for Workgroups 3.11.
- 386 Enhanced mode, intended for newer computers with a Intel 80386 processor or above, and corresponding to its protected mode and virtual 8086 mode. This implemented all the benefits of Standard mode, plus 32-bit addressing and paging for faster memory access, plus virtual 8086 mode for safer execution of MS-DOS programs: each of them now ran in a virtual machine. In the previous modes, multiple MS-DOS programs could only be run in full-screen, and only the program currently active was executing; but in 386 enhanced mode, they could be run simultaneously in separate windows.

Windows 3.0 also provided multimedia extensions, providing support for sound cards and CD-Roms. Windows also allowed a user to better multitask older MS-DOS based software compared to Windows/386, thanks to the introduction of virtual memory.

Windows 3.1 and 3.11 Base and Workgroup – 1992-1993

Windows 3.1 contained several minor upgrades, such as True Type fonts, but provided no changes to the architecture. Windows for Workgroups provided the Windows 3.x base layered with improved network drivers and protocol stacks, and support for peer-to-peer networking. One optional download for WfW was the 'Wolverine' TCP/IP protocol stack, which allowed for easy access to the Internet through corporate networks (http://en.wikipedia.org/wiki/History_of_Microsoft_Windows, 2006).

Figure 7: Windows 3.x desktop and applications (http://en.wikipedia.org/wiki/Windows_3.1x)

Windows 95 – August 1995

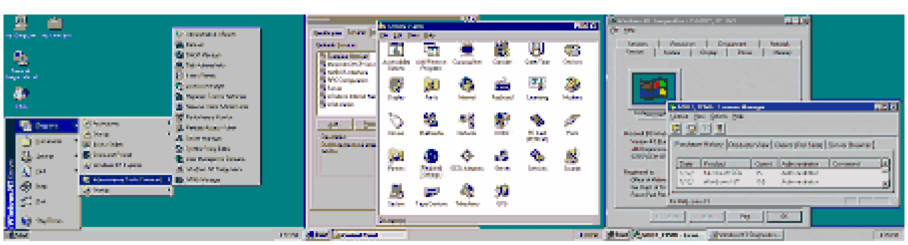
Windows 95 is a consumer-oriented graphical user interface-based operating system and was a significant progression from the company's previous versions of Windows. Windows 95 was intended to combine the functions of Microsoft's formerly separate MS-DOS and Windows products. It featured significant improvements over the popular Windows 3.1, most visibly the graphical user interface (GUI).

Figure 8: Windows 95 Desktop and applications (http://en.wikipedia.org/wiki/Windows_95, 2006).

There were also large changes to the underlying workings, including support for 255-character mixed-case long filenames and preemptively multitasked protected-mode 32-bit applications. Windows 95 also provided for 32-bit file access, which was necessary for the long file names feature introduced with Windows 95 through the use of the VFAT file system. It was available to both Windows programs and MS-DOS programs started from Windows (they had to be adapted slightly, since accessing long file names required using larger pathname buffers and hence different system calls) (http://en.wikipedia.org/wiki/Windows_95, 2006).

Windows New Technology (NT) 4.0 – 1996

Windows NT is a 32-bit Windows system available in workstation and server editions with a graphical environment similar to Windows 95. While more stable than Windows 95, it is also less flexible from a desktop perspective. Much of the stability is gained by virtualising the hardware and having software applications access the system APIs rather than the hardware directly as was done in DOS. Another set of important features also were Microsoft Transaction Server for network applications, and Microsoft Message Queuing (MSMQ), which improved communication. Also, one significant change from previous versions of Windows NT was that the Graphics Device Interface (GDI) was incorporated into kernel to speed up the GUI, which was carried forward in future windows versions (http://en.wikipedia.org/wiki/Windows_NT_4.0, 2006).



Figures 9, 10, 11: Windows NT 4.0 desktop, control panel, and networking (http://en.wikipedia.org/wiki/Windows_NT_4.0, 2006).

Windows 98 – June 1998

The new operating system was essentially an updated version of Windows 95, and like that earlier version, it was a hybrid 16-bit/32-bit monolithic product. Among its features were better AGP support, functional USB drivers, and support for multiple monitors and WebTV. Also added was support for the FAT32 file system which allowed Windows 98 support disk partitions larger than the two gigabyte maximum accepted by Windows 95. As in later releases of Windows 95, Internet Explorer continued to be integrated into the Windows Explorer interface (a feature which was called Active Desktop). It was also the first version of Windows to support Advanced Configuration and Power Interface (ACPI) (http://en.wikipedia.org/wiki/Windows_98, 2006).

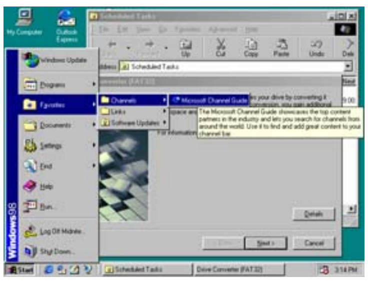
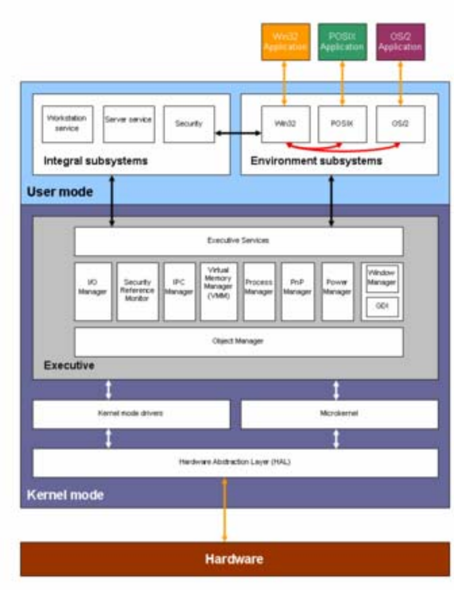


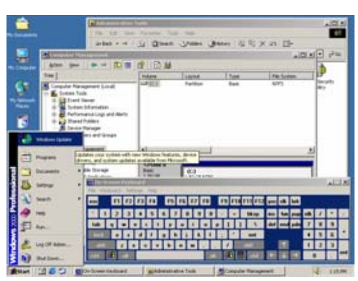
Figure 12: Windows 98 desktop and applications (http://en.wikipedia.org/wiki/Windows_98, 2006).

Windows 2000 – February 2000

Windows 2000 is a preemptible and interruptible, graphical, business-oriented operating system that was designed to work with either uniprocessor or symmetric multi-processor (SMP) 32-bit Intel x86 computers. Windows 2000 is classified as a hybrid-kernel operating system, and its architecture is divided into two modes: user mode and kernel mode. The kernel mode provides unrestricted access to system resources and facilitates the user mode, which is heavily restricted and designed for most applications. Windows 2000 introduced the Microsoft Management Console (MMC), which is used to create, save, and open administrative tools. (http://en.wikipedia.org/wiki/Windows_2000, 2006).



Figures 13, and 14: The Windows 2000 operating system architecture consists of two layers (user mode and kernel mode), with many different modules within both of these layers. (left); on-screen keyboard accessibility feature and desktop with apps (below) (http://en.wikipedia.org/wiki/Windows_2000, 2006)



Windows ME – September 2000

Windows Millennium Edition (ME) was touted to be the "home version" of 2000 and a "pre-version" of XP. The only notable features of ME were: Windows Me is an MS-DOS based version just like Windows 95 and Windows 98 but with access to real mode MS-DOS restricted for faster system boot time; Windows Me introduced the "System Restore" logging and reversion system, which was meant to simplify troubleshooting and solving problems. File protection, introduced in Windows 2000, was also brought into ME (http://en.wikipedia.org/wiki/Windows_Me, 2006). Windows ME was short lived, but paved the way for XP.

Windows XP and Beyond

Windows XP, released in 2001, is the successor to both Windows 2000 and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT kernel and architecture. With the release of Windows XP the Windows 95/98 architecture was finally discontinued. Windows XP is known for its improved stability and efficiency over previous versions of Windows. It presents a significantly redesigned graphical user interface (GUI), a change Microsoft promoted as more user-friendly than previous versions of Windows. New software management capabilities were introduced to avoid the "DLL hell" that plagued older consumer versions of Windows. Windows XP analyzes the performance impact of visual effects and uses this to determine whether to enable them, so as to prevent the new functionality from consuming excessive additional processing overhead. Users can further customize these settings (http://en.wikipedia.org/wiki/Windows_XP, 2006).

Figure 15: Windows XP desktop
(http://en.wikipedia.org/wiki/Windows_XP, 2006

Windows XP has combined both power and ease of use. As Windows evolved through the versions to XP, the WIMP concept became more powerful, expandable, and more feature rich. Windows XP also provided better support for speech to text interface, as well as optical character recognition (OCR). Both forms of interaction will further change how users interact with future versions of windows.

Windows 2003 – April 2003

Windows Server 2003 (which carries the version number NT 5.2) is the follow-up to Windows 2000 Server, incorporating compatibility and other features from Windows XP. Unlike Windows 2000 Server, Windows Server 2003's default installation has none of the server components enabled, to reduce the attack surface of new machines. Windows Server 2003 includes compatibility modes to allow older applications to run with greater stability. It was made more compatible with Windows NT 4.0 domain-based networking. Incorporating and upgrading a Windows NT 4.0 domain to Windows 2000 was considered difficult and time consuming, and generally was considered an all or nothing upgrade particularly when dealing with Active Directory. Windows Server 2003 brought in enhanced Active Directory compatibility, and better deployment support, to ease the transition from Windows NT 4.0 to Windows Server 2003 and Windows XP Professional (http://en.wikipedia.org/wiki/Windows_Server_2003, 2006).

Windows Vista – Forthcoming

According to Microsoft, Windows Vista has hundreds of new features, such as an updated graphical user interface and visual style dubbed Windows Aero (an acronym for Authentic, Energetic, Reflective, and Open), improved searching features, new multimedia creation tools, and completely redesigned networking, audio, print and display sub-systems. Microsoft claims that Vista also aims to increase the level of communication between machines on a home network using peer-to-peer technology, making it easier to share files, password settings, and digital media between computers and devices. For developers, Vista introduces version 3.0 of the .NET Framework, which aims to make it significantly easier for developers to write high-quality applications than with the traditional Windows API. High-end editions of Vista are expected to include Microsoft's Virtual PC, so that previous versions of Windows can be run simultaneously with Windows Vista on the same machine, with a view to running applications incompatible with Windows Vista.

Figure 16: Windows Vista proposed interface (http://en.wikipedia.org/wiki/Windows_Vista, 2006).

What Will Future Versions of Windows Look Like?

Within a generation, the WIMP interface concept will become a historical relic. Pointing will still be the way to command our machines, speech recognition will eventually complement and maybe replace the “clicking” action (Brooks Jr,, 1995, p. 264). Another interface which will eventually take over Windows is Virtual Interface (VI) architecture. VI Architecture promises to improve performance for distributed applications, especially clustered and tiered database applications. These improvements result from a network adapter and driver architecture that eliminates much kernel-mode processing. VI Architecture support can benefit application performance in several ways. First, the reduced instruction path length means faster I/O completion. Second, using fewer CPU cycles for network I/O leaves more CPU cycles free for applications on the server—a potentially significant advantage for network I/O-intensive applications (Green, 2000).

Things to Consider

Future versions will be targeted directly at revolutionizing the way we interact with our home and office PCs. For instance, the "Start" philosophy, introduced in Windows 95, may be completely replaced by the "new interface" . The Explorer shell will be replaced in its entirety, with features such as the taskbar being replaced by a new concept based on the last 10 years of R&D at the Microsoft "VIBE" research lab. Projects such as GroupBar and LayoutBar are expected to make an appearance, allowing users to more effectively manage and keep track of their applications and documents while in use, and a new way of launching applications is expected—among other ideas, Microsoft is investigating a pie menu-type circular interface, similar in function to the dock in Mac OS X (http://en.wikipedia.org/wiki/Windows_Vienna, 2006).

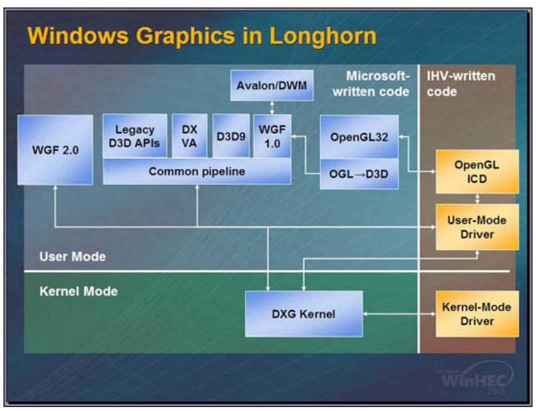


Figure 17: Is this the future? Windows Graphics structure in the “elusive” Longhorn release (Case, 2005)

Virtualization should be another thing to consider. Virtualization allows multiple applications to "time-slice" the graphics and other hardware and extends to the idea of cross-process shared surfaces—two applications running different surfaces can share surface data. The scheduler manages the command stream traffic to the hardware (Case, 2005).

The GUI in every version of Windows, including Windows XP, has been a single-buffered display. One adverse effect of this design has been animation-tearing when windows were moved around on screen. In Longhorn, an image of the desktop lives in the back buffer, and moving windows around will no longer result in windows

Figure 18: Desktop Window Manager architecture in the future (Case, 2005)

tearing or streaking. These copies are essentially textures, and any texture operation that the hardware can deliver can be applied to a window (Case, 2005). Full-screen exclusive mode is now abstracted up one layer.
The app that needs full-screen exclusive (a game, for example) can get exclusive access to a display output context. The Desktop Window Manager disappears when this happens. The outcome of this is that if you have two monitors, the two displays are now independent. So a game might run on one monitor, the second display could still have desktop window manager (DWM) running. Additionally, the new concept of Windows Graphic Foundation will change the graphical architecture of the GUI and provide better hardware consistency. Graphical Processor Units (GPUs – found in video cards) will become similar to CPUs; the architecture and performance characteristics may vary widely, but the same programs will ultimately deliver the same results (Case, 2005).

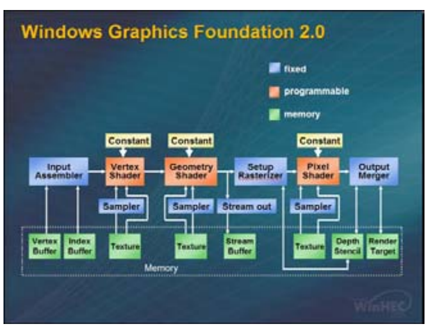


Figure 20: WGF in future Windows (Case, 2005)

The future of Windows will be nothing short of a complete overhaul of Windows as we know it today. Massively rebuilt GUIs, better user interaction, further evolution of touchscreens, application services, and the ever evolving dotNet framework will all play a role in the future of Windows.

References

Brooks, Jr., F. P. (1995). *The Mythical Man-Month (Anniversary ed..).* Reading, MA: Addison-Wesley.

Case, L. (2005). *The Future of Windows Graphics Technology*. Retrieved on July 23, 2006 from http://www.extremetech.com/article2/0,1558,1791702,00.asp

Coustan, D. & Franklin, C. (2006). *How Operating Systems Work*. Retrieved on July 22, 2006 from http://www.howstuffworks.com/operating-system1.htm

Green, J. (2000, Nov.). Visual Interface Architecture. *Windows IT Pro.* Retrieved on July 18, 2006 from http://www.windowsitpro.com/Windows/issues/issueID/427/427.htm

Tanenbaum, A. (2006). *Structured Computer Organization (5th ed.).* Upper Saddle River, NJ: Pearson Prentice Hall.

Wikipedia web site. History of Microsoft Windows. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/History_of_Microsoft_Windows
.

Wikipedia web site. Windows 1.0. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_1.0

Wikipedia web site. Windows 2.0. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_2.0

Wikipedia web site. Windows 3.0. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_3.0

Wikipedia web site. Windows 3.1x. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_3.1x

Wikipedia web site. Windows 95. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_95

Wikipedia web site. Windows NT 4.0. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_NT_4.0

Wikipedia web site. Windows 98. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_98

Wikipedia web site. Windows ME. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_Me

Wikipedia web site. Windows XP. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_XP

Wikipedia web site. Windows Server 2003. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_Server_2003

Wikipedia web site. Windows Vista. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_Vista

Wikipedia web site. Windows Vienna and Beyond. Retrieved July 16, 2006 from: http://en.wikipedia.org/wiki/Windows_Vienna