CHAPTER 22

Windows XP

The Microsoft Windows XP operating system is a 32/64-bit preemptive multitasking operating system for AMD K6/K7, Intel IA32/IA64 and later microprocessors. The successor to Windows NT/2000, Windows XP, is also intended to replace the MS-DOS operating system. Key goals for the system are security, reliability, ease of use, Windows and POSIX application compatibility, high performance, extensibility, portability and international support. This chapter discusses the key goals for Windows XP, the layered architecture of the system that makes it so easy to use, the file system, networks, and the programming interface. Windows XP serves as an excellent case study as an example operating system.

22.1 (1) As the hardware powers on, the BIOS begins executing from ROM and loads and executes the bootstrap loader from the disk. (2) The NTLDR program is loaded from the root directory of the identified system device and determines which boot device contains the operating system. (3) NTLDR loads the HAL library, kernel, and system hive. The system hive indicates the required boot drivers and loads them. (4) Kernel execution begins by initializing the system and creating two processes: the system process containing all internal worker threads, and the first user-mode initialization process: SMSS. (5) SMSS further initializes the system by establishing paging files and loading device drivers. (6) SMSS creates two processes: WINLOGON, which brings up the rest of the system, and CSRSS (the Win32 subsystem process).

22.2 A process is an executing instance of an application containing one or more threads. Threads are the units of code that are scheduled by the operating system. A process is started when some other process calls the CreateProcess routine, which loads any dynamic link libraries used by the process, resulting in a primary thread. Additional threads can also be created. Each thread is created with its own stack with a wrapper function providing thread synchronization.

22.3 (1) Virtual memory provides several functions that allow an application to reserve and release memory, specifying the virtual address at which the memory is allocated. (2) A file may be memory-mapped into address space, providing a means for two processes to share memory. (3) When a Win32 process is initialized, it is created with a default heap. Private heaps can be created that provide regions of reserved address space for applications. Thread management functions are provided to allocate and control thread access to private heaps. (4) A thread-local storage mechanism provides a way for global and static data to work properly in a multithreaded environment. Thread-lock storage allocates global storage on a per-thread basis.

22.4 When a process accesses a no-access page, an exception is raised. This feature is used to check whether a faulty program accesses beyond the end of an array. The array needs to be allocated in a manner such that it appears at the end of a page, so that buffer overruns would cause exceptions.

22.5 Deferred procedure calls are used to postpone interrupt processing in situations where the processing of device interrupts can be broken into a critical portion that is used to unblock the device and a non-critical portion that can be scheduled later at a lower priority. The non-critical section of code is scheduled for later execution by queuing a deferred procedure call.

22.6 (1) The HAL (Hardware Abstraction Layer) creates operating system portability by hiding hardware differences from the upper layers of the operating system. Administrative details of low-level facilities are provided by HAL interfaces. HAL presents a virtual-machine interface that is used by the kernel dispatcher, the executive and device drivers. (2) The kernel layer provides a foundation for the executive functions and user-mode subsystems. The kernel remains in memory and is never preempted. Its responsibilities are thread scheduling, interrupt and exception handling, low-level processor synchronization, and power failure recovery. (3) The executive layer provides a set of services used by all subsystems: object manager, virtual memory manager, process manager, local procedure call facility, I/O manager, security monitor, plug-and-play manager, registry, and booting.

22.7 Windows16 execution environment provides a virtual environment for executing 16-bit applications that use the Windows 3.1 kernel interface. The interface is supported in software using stub routines that call the appropriate Win32 API subroutines by converting 16-bit addresses into 32-bit addresses. This allows the system to run legacy applications. The environment can multitask with other processes on Windows XP. It can contain multiple Windows16 applications, but all applications share the same address space and the same input queue. Also, one can execute only one Windows16 application at a given point in time. A Windows16 application can therefore crash other Windows16 applications by corrupting the address space, but it cannot corrupt the address spaces of Win32 applications. Multiple Windows16 execution environments could also coexist.

22.8 A fiber is a sequential stream of execution within a process. A process can have multiple fibers in it, but unlike threads, only one fiber at a time is permitted to execute. The fiber mechanism is used to support legacy applications written for a fiber-execution model.

22.9 Environmental subsystems are user-mode processes layered over the native executable services to enable Windows XP to run programs developed for other operating systems. (1) A Win32 application called the virtual DOS machine (VDM) is provided as a user-mode process to run MS-DOS applications. The VDM can execute or emulate Intel 486 instructions and also provides routines to emulate MS-DOS BIOS services and provides virtual drivers for screen, keyboard, and communication ports. (2) Windows-on-windows (WOW32) provides kernel and stub routines for Windows 3.1 functions. The stub routines call the appropriate Win32 subroutines, converting the 16-bit addresses into 32-bit addresses.

22.10 The VM manager uses a page-based management scheme. Pages of data allocated to a process that are not in physical memory are either stored in paging files on disk or mapped to a regular file on a local or remote file system. To improve performance of this scheme, a privileged process is allowed to lock selected pages in physical memory preventing those pages from being paged out. Furthermore, since when a page is used, adjacent pages will likely be used in the near future, adjacent pages are prefetched to reduce the total number of page faults.

22.11 Data is communicated using one of the following three facilities: 1) messages are simply copied from one process to the other, 2) a shared memory segment is created and messages simply contain a pointer into the shared memory segment, thereby avoiding copies between processes, 3) a process directly writes into the other process's virtual space.

22.12 In NTFS, all file-system data structure updates are performed inside transactions. Before a data structure is altered, the transaction writes a log record containing redo and undo information. A commit record is written to the log after a transaction has succeeded. After a crash the file system can be restored to a consistent state by processing the log records, first redoing operations for committed transactions and undoing operations for transactions that did not successfully commit. This scheme does not guarantee that user file contents are correct after a recovery, but rather that the file-system data structures (file metadata) are undamaged and reflect some consistent state that existed before the crash.

22.13 Environmental subsystems are user-mode processes layered over the native executable services to enable Windows XP to run programs developed for other operating systems. (1) A Win32 application called the virtual DOS machine (VDM) is provided as a user-mode process to run MS-DOS applications. The VDM can execute or emulate Intel 486 instructions and also provides routines to emulate MSDOS BIOS services and provides virtual drivers for screen, keyboard, and communication ports. (2) Windows-on-Windows (WOW32) provides kernel and stub routines for Windows 3.1 functions. The stub routines call the appropriate Win32 subroutines, converting the 16-bit addresses into 32-bit addresses.

22.14 Objects present a generic set of kernel mode interfaces to user-mode programs. Objects are manipulated by the executive-layer object manager. The job of the object manager is to supervise the allocation and use of all managed objects.

22.15 Each page table entry is 64 bits wide and each page is 8 KB on the IA64. Consequently, each page can contain 1024 page table entries. The virtual memory system therefore requires three levels of page tables to translate a virtual address to a physical address in order to address a 8-TBvirtual address space. (The first-level page table is indexed using the first 10 bits of the virtual address, the second-level page table using the next 10 bits, and the third-level page table is indexed using the next 10 bits, with the remaining 13 bits used to index into the page.) If the virtual address space is bigger, more levels would be required in the page table organization, and therefore more memory references would be required to translate a virtual address to the corresponding physical address during a TLB fault. The decision regarding the 43-bit address space represents a trade off between the size of the virtual address space and the cost of performing an address translation.

22.16 The NTFS namespace is organized as a hierarchy of directories where each directory uses a B+ tree data structure to store an index of the filenames in that directory. The index root of a directory contains the top level of the B+ tree. Each entry in the directory contains the name and file reference of the file as well as the update timestamp and file size. The UNIX operating system simply stores a table of entries mapping names to i-node numbers in a directory file. Lookups and updates require a linear scan of the directory structure in UNIX systems.

22.17 The I/O manager is responsible for file systems, device drivers, and network drivers. The I/O manager keeps track of which device drivers, filter drivers, and file systems are loaded and manages buffers for I/O requests. It furthermore assists in providing memory-mapped file I/O and controls the cache manager for the whole I/O system.

22.18 In contrast to other operating systems where caching is done by the file system, Windows XP provides a centralized cache manager which works closely with the VM manager to provide caching services for all components under control of the I/O manager. The size of the cache changes dynamically depending upon the free memory available in the system. The cache manager maps files into the upper half of the system cache address space. This cache is divided into blocks that can each hold a memory-mapped region of a file.

22.19 User-mode code can access kernel-mode objects by using a reference value called a handle. An object handle is thus an identifier (unique to a process) that allows access and manipulation of a system resource. When a user-mode process wants to use an object it calls the object manager's open method. A reference to the object is inserted in the process's object table and a handle is returned. Processes can obtain handles by creating an object, opening an existing object, receiving a duplicated handle from another process, or inheriting a handle from a parent process.