Windows XP

Practice Exercises

22.1 What type of operating system is Windows XP? Describe two of its major features.
Answer: A 32/64 bit preemptive multitasking operating system supporting multiple users. (1) The ability automatically to repair application and operating system problems. (2) Better networking and device experience (including digital photography and video).

22.2 List the design goals of Windows XP. Describe two in detail.
Answer: Design goals include security, reliability, Windows and POSIX application compatibility, high performance, extensibility, portability and international support. (1) Reliability was perceived as a stringent requirement and included extensive driver verification, facilities for catching programming errors in user-level code, and a rigorous certification process for third-party drivers, applications, and devices. (2) Achieving high performance required examination of past problem areas such as I/O performance, server CPU bottlenecks, and the scalability of multithreaded and multiprocessor environments.

22.3 Describe the booting process for a Windows XP system.
Answer: (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).

22.4 Describe the three main architectural layers of Windows XP.
Answer: (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.

22.5 What is the job of the object manager?
Answer: 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.

22.6 What types of services does the process manager provide? What is a local procedure call?
Answer: The process manager provides services for creating, deleting, and using processes, threads and jobs. The process manager also implements queuing and delivery of asynchronous procedure calls to threads. The local procedure call (LPC) is a message-passing system. The operating system uses the LPC to pass requests and results between client and server processes within a single machine, in particular between Windows XP subsystems.

22.7 What are the responsibilities of the I/O manager?
Answer: 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.

22.8 Does Windows XP offer any user-mode processes that enable it to run programs developed for other operating systems? Describe two of these subsystems.
Answer: 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.

22.9 What types of networking does Windows XP support? How does Windows XP implement transport protocols? Describe two networking protocols.
Answer: Support is provided for both peer-to-peer and client-server networking. Transport protocols are implemented as drivers. (1) The TCP/IP package includes SNMP, DHCP, WINS, and NetBIOS support. (2) Point-to-point tunneling protocol is provided to communicate between remote-access modules running on Windows XP servers and other client systems connected over the internet. Using this scheme, multi-protocol virtual private networks (VPNs) are supported over the internet.

22.10 How is the NTFS namespace organized? Describe.
Answer: 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 file names 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.

22.11 How does NTFS handle data structures? How does NTFS recover from a system crash? What is guaranteed after a recovery takes place?
Answer: 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.

22.12 How does Windows XP allocate user memory?
Answer: User memory can be allocated according to several schemes: virtual memory, memory-mapped files, heaps, and thread-local storage.

22.13 Describe some of the ways an application can use memory via the Win32 API.
Answer: (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.