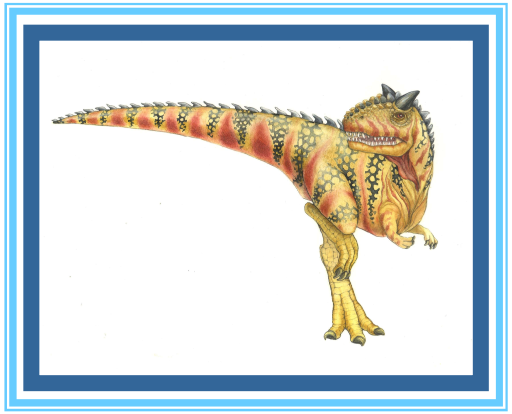
Chapter 10: Windows 10



Chapter 10: Windows

- History
- Design Principles
- System Components
- Environmental Subsystems
- Networking
- Programmer Interface

Objectives

- To explore the principles upon which Windows 10 is designed and the specific components involved in the system
- To understand how Windows 10 can run programs designed for other operating systems

Windows 10

- 32-bit preemptive multitasking operating system for Intel microprocessors
- Key goals for the system:
  - portability
  - security
  - POSIX compliance
  - multiprocessor support
  - extensibility
  - international support
  - compatibility with MS-DOS and MS-Windows applications.
- Uses a micro-kernel architecture
- Available in six client versions, Starter, Home Basic, Home Premium, Professional, Enterprise and Ultimate. With the exception of Starter edition (32-bit only) all are available in both 32-bit and 64-bit.
- Available in three server versions (all 64-bit only), Standard, Enterprise and Datacenter

History

- In 1988, Microsoft decided to develop a “new technology” (NT) portable operating system that supported both the OS/2 and POSIX APIs
- Originally, NT was supposed to use the OS/2 API as its native environment but during development NT was changed to use the Win32 API, reflecting the popularity of Windows 3.0.

Design Principles

- Extensibility — layered architecture
  - Executive, which runs in protected mode, provides the basic system services
  - On top of the executive, several server subsystems operate in user mode
  - Modular structure allows additional environmental subsystems to be added without affecting the executive
- Portability — Windows 10 can be moved from one hardware architecture to another with relatively few changes
  - Written in C and C++
  - Processor-specific portions are written in assembly language for a given processor architecture (small amount of such code).
  - Platform-dependent code is isolated in a dynamic link library (DLL) called the “hardware abstraction layer” (HAL)

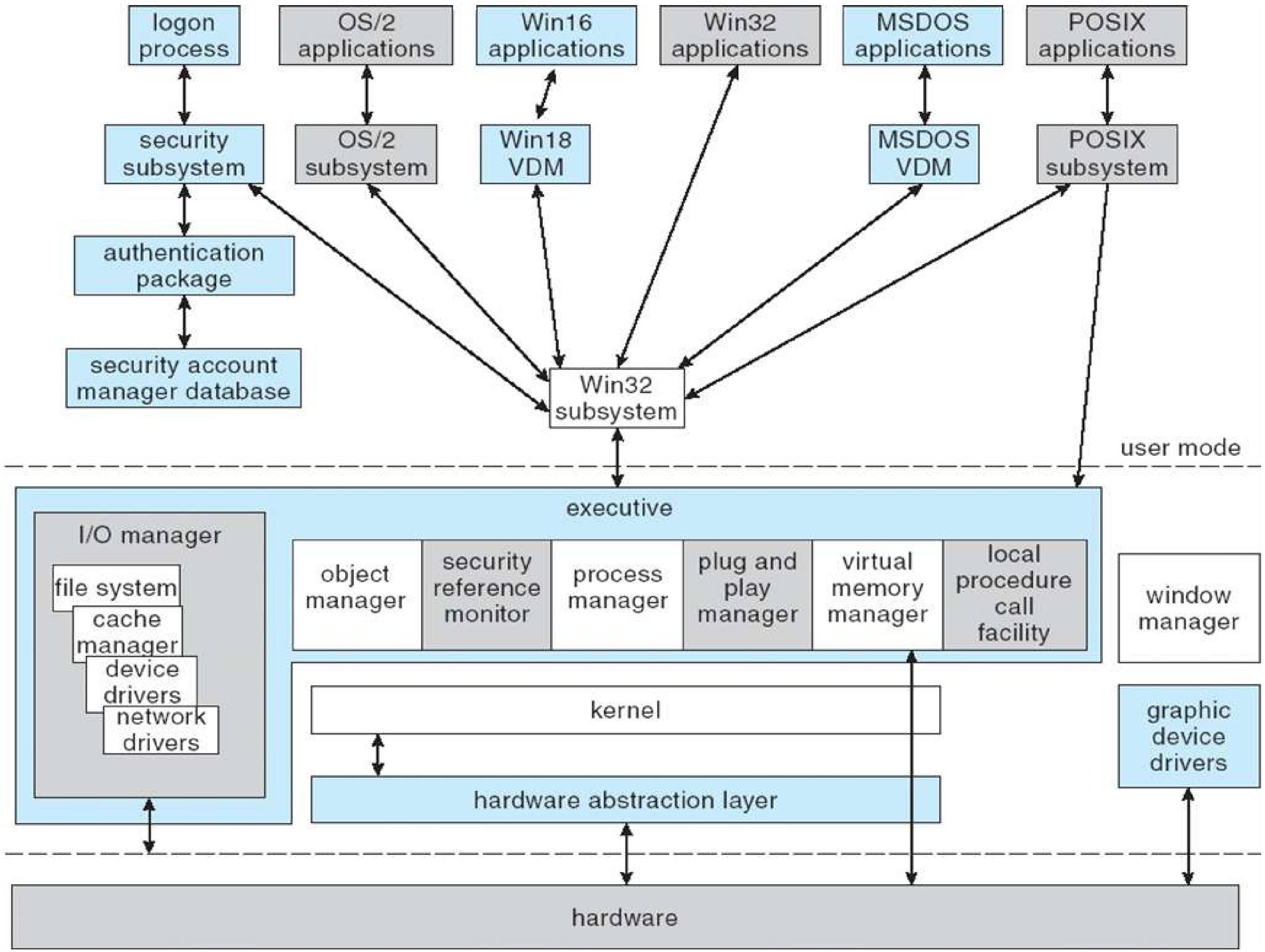
Design Principles (Cont.)

- Reliability — Windows 10 uses hardware protection for virtual memory, and software protection mechanisms for operating system resources
- Compatibility — applications that follow the IEEE 1003.1 (POSIX) standard can be complied to run on 10 without changing the source code
- Performance — Windows 10 subsystems can communicate with one another via high-performance message passing
  - Preemption of low priority threads enables the system to respond quickly to external events
  - Designed for symmetrical multiprocessing
- International support — supports different locales via the national language support (NLS) API

Windows 10 Architecture

- Layered system of module
- Protected mode — **hardware abstraction layer** (**HAL**), kernel, executive
- User mode — collection of subsystems
  - Environmental subsystems emulate different operating systems
  - Protection subsystems provide security functions

Depiction of 10 Architecture



System Components — Kernel

- Foundation for the executive and the subsystems
- Never paged out of memory; execution is never preempted
- Four main responsibilities:
  - thread scheduling
  - interrupt and exception handling
  - low-level processor synchronization
  - recovery after a power failure
- Kernel is object-oriented, uses two sets of objects
  - *dispatcher objects* control dispatching and synchronization (events, mutants, mutexes, semaphores, threads and timers)
  - *control objects* (asynchronous procedure calls, interrupts, power notify, power status, process and profile objects)

Kernel — Process and Threads

- The process has a virtual memory address space, information (such as a base priority), and an affinity for one or more processors.
- Threads are the unit of execution scheduled by the kernel’s dispatcher.
- Each thread has its own state, including a priority, processor affinity, and accounting information.
- A thread can be one of six states: *ready, standby, running, waiting, transition*, and *terminated.*

Kernel — Scheduling

- The dispatcher uses a 32-level priority scheme to determine the order of thread execution.
  - Priorities are divided into two classes
    - The real-time class contains threads with priorities ranging from 16 to 31
    - The variable class contains threads having priorities from 0 to 15
- Characteristics of Windows 7's priority strategy
  - Trends to give very good response times to interactive threads that are using the mouse and windows
  - Enables I/O-bound threads to keep the I/O devices busy
  - Complete-bound threads soak up the spare CPU cycles in the background

Kernel — Scheduling (Cont.)

- Scheduling can occur when a thread enters the ready or wait state, when a thread terminates, or when an application changes a thread's priority or processor affinity
- Real-time threads are given preferential access to the CPU; but 7 does not guarantee that a real-time thread will start to execute within any particular time limit .
  - This is known as *soft realtime.*

Windows 7 Interrupt Request Levels

interrupt levels	types of interrupts
31	machine check or bus error
30	power fail
29	interprocessor notification (request another processor to act; e.g., dispatch a process or update the TLB)
28	clock (used to keep track of time)
27	profile
3–26	traditional PC IRQ hardware interrupts
2	dispatch and deferred procedure call (DPC) (kernel)
1	asynchronous procedure call (APC)
0	passive

Kernel — Trap Handling

- The kernel provides trap handling when exceptions and interrupts are generated by hardware of software.
- Exceptions that cannot be handled by the trap handler are handled by the kernel's **exception dispatcher**.
- The interrupt dispatcher in the kernel handles interrupts by calling either an interrupt service routine (such as in a device driver) or an internal kernel routine.
- The kernel uses spin locks that reside in global memory to achieve multiprocessor mutual exclusion.

Executive — Object Manager

- Windows 7 uses objects for all its services and entities; the object manger supervises the use of all the objects
  - Generates an object *handle*
  - Checks security
  - Keeps track of which processes are using each object
- Objects are manipulated by a standard set of methods, namely `create`, `open`, `close`, `delete`, `query name`, `parse` and `security`.

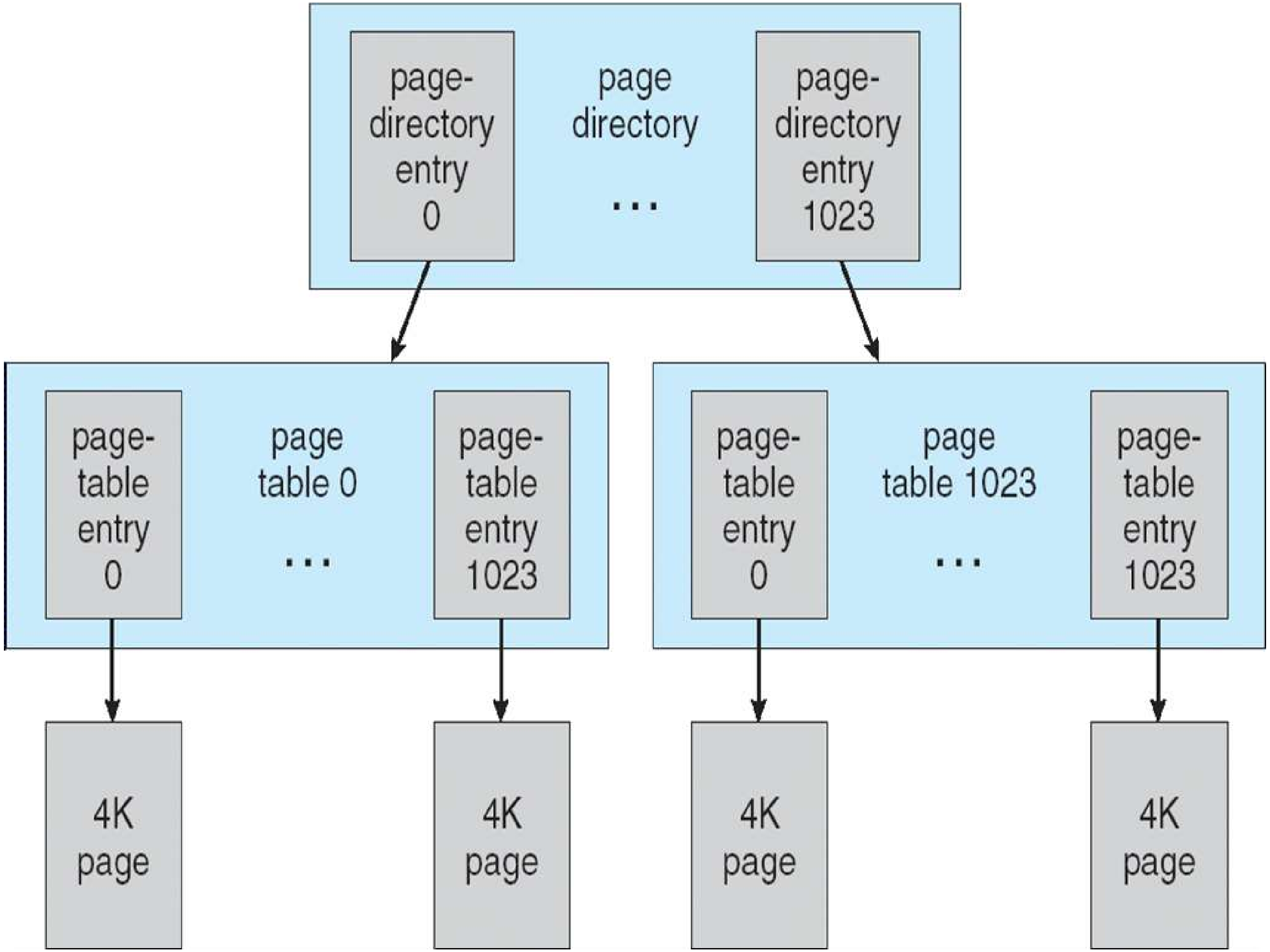
Executive — Naming Objects

- The Windows 7 executive allows almost any object to be given a name, which may be either permanent or temporary. Exceptions are process, thread and some others object types.
- Object names are structured like file path names in MS-DOS and UNIX.
- Windows 7 implements a *symbolic link object*, which is similar to *symbolic links* in UNIX that allow multiple nicknames or aliases to refer to the same file.
- A process gets an object handle by creating an object by opening an existing one, by receiving a duplicated handle from another process, or by inheriting a handle from a parent process.
- Each object is protected by an access control list.

Executive — Virtual Memory Manager

- The design of the VM manager assumes that the underlying hardware supports virtual to physical mapping a paging mechanism, transparent cache coherence on multiprocessor systems, and virtual addressing aliasing.
- The VM manager in Windows 7 uses a page-based management scheme with a page size of 4 KB.
- The Windows 7 VM manager uses a two step process to allocate memory
  - The first step reserves a portion of the process's address space
  - The second step commits the allocation by assigning space in the system's paging file(s)

Virtual-Memory Layout

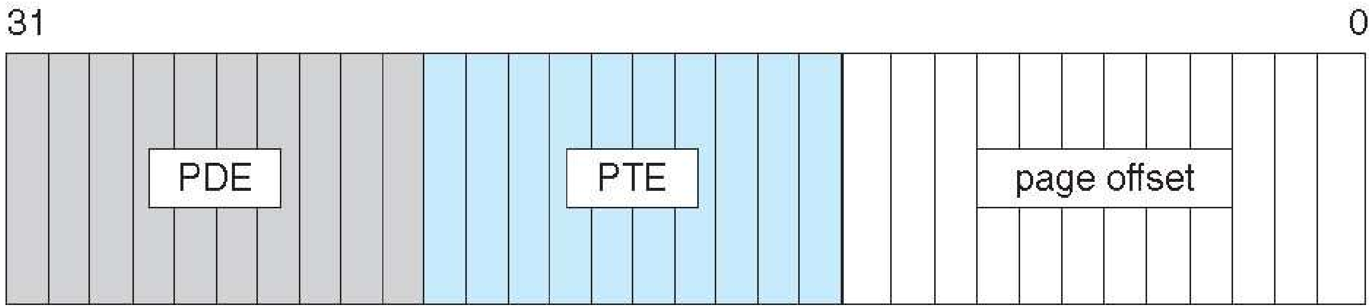


Virtual Memory Manager (Cont.)

- The virtual address translation in Windows 7 uses several data structures
  - Each process has a *page directory* that contains 1024 *page directory entries* of size 4 bytes.
  - Each page directory entry points to a *page table* which contains 1024 *page table entries* (PTEs) of size 4 bytes.
  - Each PTE points to a 4 KB *page frame* in physical memory.
- A 10-bit integer can represent all the values form 0 to 1023, therefore, can select any entry in the page directory, or in a page table.
- This property is used when translating a virtual address pointer to a bye address in physical memory.
- A page can be in one of six states: valid, zeroed, free standby, modified and bad.

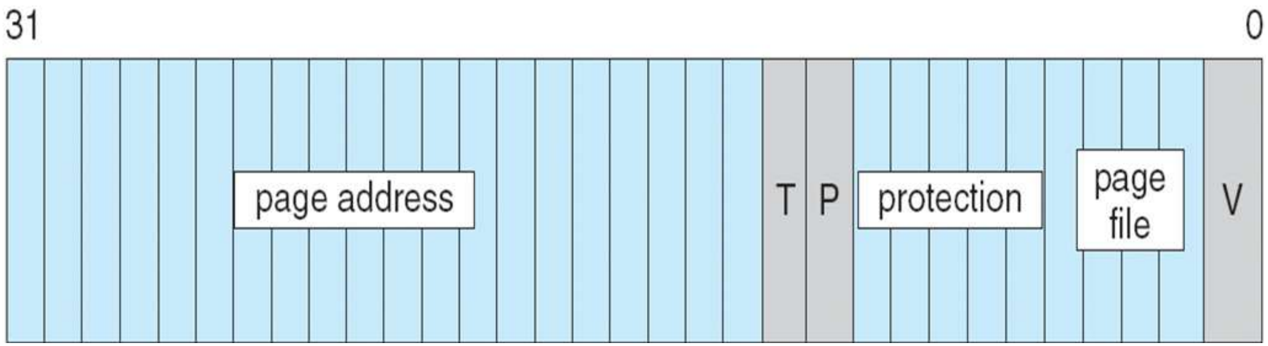
Virtual-to-Physical Address Translation

- 10 bits for page directory entry, 20 bits for page table entry, and 12 bits for byte offset in page



PDE: Page Directory Entry
PTE: Page Table Entry

Page File Page-Table Entry



5 bits for page protection, 20 bits for page frame address, 4 bits to select a paging file, and 3 bits that describe the page state. V = 0

Executive — Process Manager

- Provides services for creating, deleting, and using threads and processes
- Issues such as parent/child relationships or process hierarchies are left to the particular environmental subsystem that owns the process.

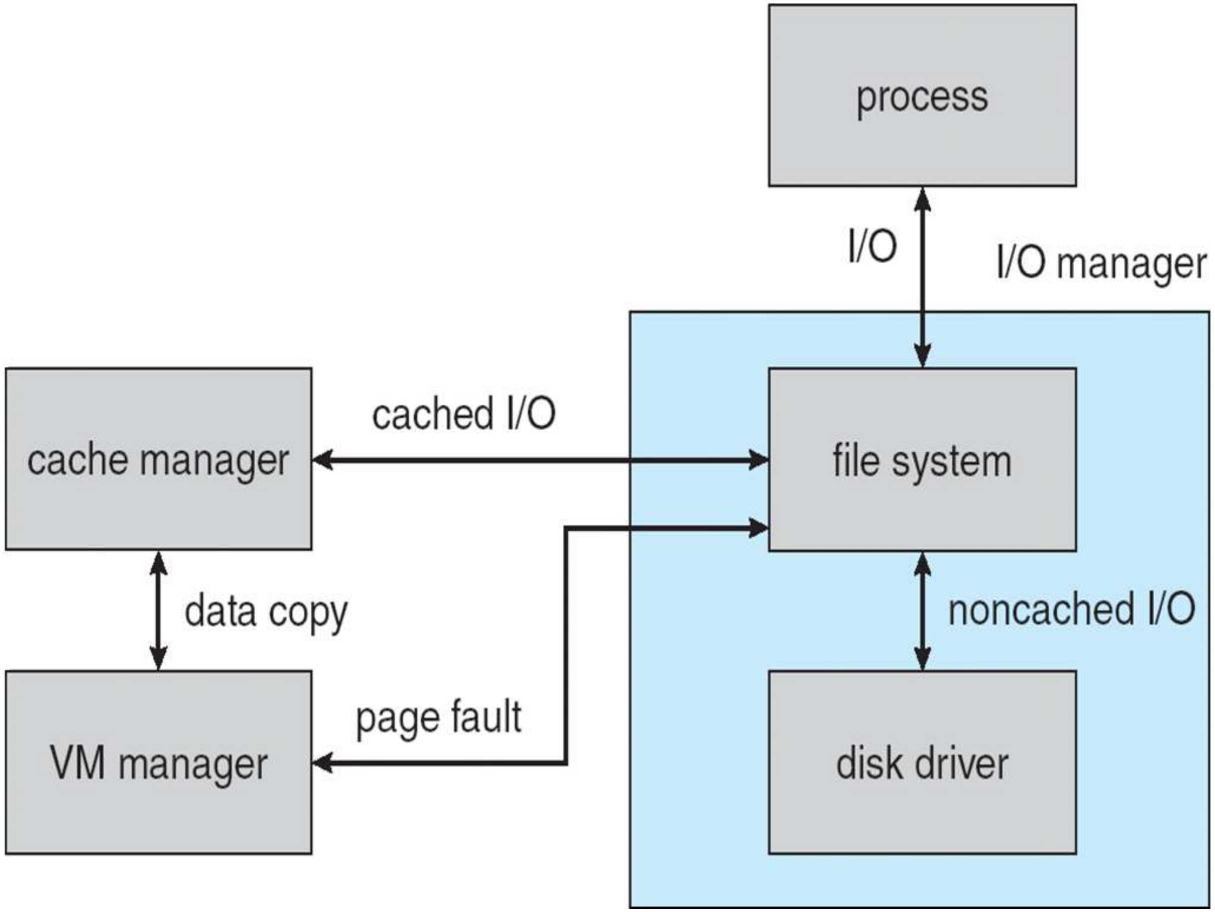
Executive — Local Procedure Call Facility

- The LPC passes requests and results between client and server processes within a single machine.
- In particular, it is used to request services from the various Windows 7 subsystems.
- When a LPC channel is created, one of three types of message passing techniques must be specified.
  - First type is suitable for small messages, up to 256 bytes; port's message queue is used as intermediate storage, and the messages are copied from one process to the other.
  - Second type avoids copying large messages by pointing to a shared memory section object created for the channel.
  - Third method, called *quick* LPC was used by graphical display portions of the Win32 subsystem.

Executive — I/O Manager

- The I/O manager is responsible for
  - file systems
  - cache management
  - device drivers
  - network drivers
- Keeps track of which installable file systems are loaded, and manages buffers for I/O requests
- Works with VM Manager to provide memory-mapped file I/O
- Controls the Windows 7 cache manager, which handles caching for the entire I/O system
- Supports both synchronous and asynchronous operations, provides time outs for drivers, and has mechanisms for one driver to call another

File I/O


process
I/O
I/O manager
cache manager
cached I/O
file system
data copy
noncached I/O
page fault
VM manager
disk driver

Executive — Security Reference Monitor

- The object-oriented nature of Windows 7 enables the use of a uniform mechanism to perform runtime access validation and audit checks for every entity in the system.
- Whenever a process opens a handle to an object, the security reference monitor checks the process's security token and the object's access control list to see whether the process has the necessary rights.

Executive – Plug-and-Play Manager

- Plug-and-Play (PnP) manager is used to recognize and adapt to changes in the hardware configuration.
- When new devices are added (for example, PCI or USB), the PnP manager loads the appropriate driver.
- The manager also keeps track of the resources used by each device.

Environmental Subsystems

- User-mode processes layered over the native Windows 7 executive services to enable 7 to run programs developed for other operating system.
- Windows 7 uses the Win32 subsystem as the main operating environment; Win32 is used to start all processes.
  - It also provides all the keyboard, mouse and graphical display capabilities.
- MS-DOS environment is provided by a Win32 application called the *virtual dos machine* (VDM), a user-mode process that is paged and dispatched like any other Windows 7 thread.

Environmental Subsystems (Cont.)

- 16-Bit Windows Environment:
  - Provided by a VDM that incorporates *Windows on Windows*
  - Provides the Windows 3.1 kernel routines and sub routines for window manager and GDI functions
- The POSIX subsystem is designed to run POSIX applications following the POSIX.1 standard which is based on the UNIX model.

Environmental Subsystems (Cont.)

- OS/2 subsystems runs OS/2 applications
- Logon and Security Subsystems authenticates users logging on to Windows 7 systems
  - Users are required to have account names and passwords.
  - The authentication package authenticates users whenever they attempt to access an object in the system.
  - Windows 7 uses Kerberos as the default authentication package

File System

- The fundamental structure of the Windows 7 file system (NTFS) is a *volume*
  - Created by the Windows 7 disk administrator utility
  - Based on a logical disk partition
  - May occupy a portions of a disk, an entire disk, or span across several disks
- All *metadata*, such as information about the volume, is stored in a regular file
- NTFS uses *clusters* as the underlying unit of disk allocation
  - A cluster is a number of disk sectors that is a power of two
  - Because the cluster size is smaller than for the 16-bit FAT file system, the amount of internal fragmentation is reduced

Networking

- Windows 7 supports both peer-to-peer and client/server networking; it also has facilities for network management.
- To describe networking in Windows 7, we refer to two of the internal networking interfaces:
  - NDIS (Network Device Interface Specification) — Separates network adapters from the transport protocols so that either can be changed without affecting the other.
  - TDI (Transport Driver Interface) — Enables any session layer component to use any available transport mechanism.
- Windows 7 implements transport protocols as drivers that can be loaded and unloaded from the system dynamically.

Networking — Protocols

- The server message block (SMB) protocol is used to send I/O requests over the network. It has four message types:
  1. `Session control`
  2. `File`
  3. `Printer`
  4. `Message`
- The network basic Input/Output system (NetBIOS) is a hardware abstraction interface for networks
  - Used to:
    - Establish logical names on the network
    - Establish logical connections of sessions between two logical names on the network
    - Support reliable data transfer for a session via NetBIOS requests or *SMBs*

Networking — Protocols (Cont.)

- Windows 7 uses the TCP/IP Internet protocol version 4 and version 6 to connect to a wide variety of operating systems and hardware platforms.
- PPTP (Point-to-Point Tunneling Protocol) is used to communicate between Remote Access Server modules running on Windows 7 machines that are connected over the Internet.
- The Data Link Control protocol (DLC) is used to access IBM mainframes and HP printers that are directly connected to the network (possible on 32-bit only versions using unsigned drivers).

Networking — Dist. Processing Mechanisms

- Windows 7 supports distributed applications via named NetBIOS, named pipes and mailslots, Windows Sockets, Remote Procedure Calls (RPC), and Network Dynamic Data Exchange (NetDDE).
- NetBIOS applications can communicate over the network using TCP/IP.
- Named pipes are connection-oriented messaging mechanism that are named via the uniform naming convention (UNC).
- Mailslots are a connectionless messaging mechanism that are used for broadcast applications, such as for finding components on the network.
- Winsock, the windows sockets API, is a session-layer interface that provides a standardized interface to many transport protocols that may have different addressing schemes.

Distributed Processing Mechanisms (Cont.)

- The Windows 7 RPC mechanism follows the widely-used Distributed Computing Environment standard for RPC messages, so programs written to use Windows 7 RPCs are very portable.
  - RPC messages are sent using NetBIOS, or Winsock on TCP/IP networks, or named pipes on LAN Manager networks.
  - Windows 7 provides the Microsoft *Interface Definition Language* to describe the remote procedure names, arguments, and results.

Networking — Redirectors and Servers

- In Windows 7, an application can use the Windows 7 I/O API to access files from a remote computer as if they were local, provided that the remote computer is running an MS-NET server.
- A *redirector* is the client-side object that forwards I/O requests to remote files, where they are satisfied by a server.
- For performance and security, the redirectors and servers run in kernel mode.

Name Resolution in TCP/IP Networks

- On an IP network, name resolution is the process of converting a computer name to an IP address
  - e.g., `www.bell-labs.com` resolves to `135.104.1.14`
- Windows 7 provides several methods of name resolution:
  - Windows Internet Name Service (WINS)
  - broadcast name resolution
  - domain name system (DNS)
  - a host file
  - an LMHOSTS file

Name Resolution (Cont.)

- WINS consists of two or more WINS servers that maintain a dynamic database of name to IP address bindings, and client software to query the servers.
- WINS uses the Dynamic Host Configuration Protocol (DHCP), which automatically updates address configurations in the WINS database, without user or administrator intervention.

Programmer Interface — Access to Kernel Obj.

- A process gains access to a kernel object named `XXX` by calling the `CreateXXX` function to open a *handle* to `XXX`; the handle is unique to that process.
- A handle can be closed by calling the `CloseHandle` function; the system may delete the object if the count of processes using the object drops to 0.
- Windows 7 provides three ways to share objects between processes
  - A child process inherits a handle to the object
  - One process gives the object a name when it is created and the second process opens that name
  - `DuplicateHandle` function:
    - Given a handle to process and the handle's value a second process can get a handle to the same object, and thus share it

Programmer Interface — Process Management

- Process is started via the `CreateProcess` routine which loads any dynamic link libraries that are used by the process, and creates a *primary thread.*
- Additional threads can be created by the `CreateThread` function.
- Every dynamic link library or executable file that is loaded into the address space of a process is identified by an *instance handle.*

Process Management (Cont.)

- Scheduling in Win32 utilizes four priority classes:
  1. `IDLE_PRIORITY_CLASS` (priority level 4)
  2. `NORMAL_PRIORITY_CLASS` (level8 — typical for most processes
  3. `HIGH_PRIORITY_CLASS` (level 13)
  4. `REALTIME_PRIORITY_CLASS` (level 24)
- To provide performance levels needed for interactive programs, 7 has a special scheduling rule for processes in the `NORMAL_PRIORITY_CLASS`
  - 7 distinguishes between the *foreground process* that is currently selected on the screen, and the *background processes* that are not currently selected.
  - When a process moves into the foreground, 7 increases the scheduling quantum by some factor, typically 3.

Process Management (Cont.)

- The kernel dynamically adjusts the priority of a thread depending on whether it is I/O-bound or CPU-bound.
- To synchronize the concurrent access to shared objects by threads, the kernel provides synchronization objects, such as semaphores and mutexes
  - In addition, threads can synchronize by using the `WaitForSingleObject` or `WaitForMultipleObjects` functions.
  - Another method of synchronization in the Win32 API is the critical section.

Process Management (Cont.)

- A fiber is user-mode code that gets scheduled according to a user-defined scheduling algorithm.
  - Only one fiber at a time is permitted to execute, even on multiprocessor hardware.
  - Windows 7 includes fibers to facilitate the porting of legacy UNIX applications that are written for a fiber execution model.
- Windows 7 also introduced user-mode scheduling for 64-bit systems which allows finer grained control of scheduling work without requiring kernel transitions.

Programmer Interface — Interprocess Communication

- Win32 applications can have interprocess communication by sharing kernel objects.
- An alternate means of interprocess communications is message passing, which is particularly popular for Windows GUI applications
  - One thread sends a message to another thread or to a window.
  - A thread can also send data with the message.
- Every Win32 thread has its own input queue from which the thread receives messages.
- This is more reliable than the shared input queue of 16-bit windows, because with separate queues, one stuck application cannot block input to the other applications

Programmer Interface — Memory Management

- Virtual memory:
  - `VirtualAlloc` reserves or commits virtual memory
  - `VirtualFree` decommits or releases the memory
  - These functions enable the application to determine the virtual address at which the memory is allocated
- An application can use memory by memory mapping a file into its address space
  - Multistage process
  - Two processes share memory by mapping the same file into their virtual memory

Memory Management (Cont.)

- A heap in the Win32 environment is a region of reserved address space
  - A Win 32 process is created with a 1 MB *default heap*
  - Access is synchronized to protect the heap's space allocation data structures from damage by concurrent updates by multiple threads
- Because functions that rely on global or static data typically fail to work properly in a multithreaded environment, the thread-local storage mechanism allocates global storage on a per-thread basis
  - The mechanism provides both dynamic and static methods of creating thread-local storage

End of Chapter 10