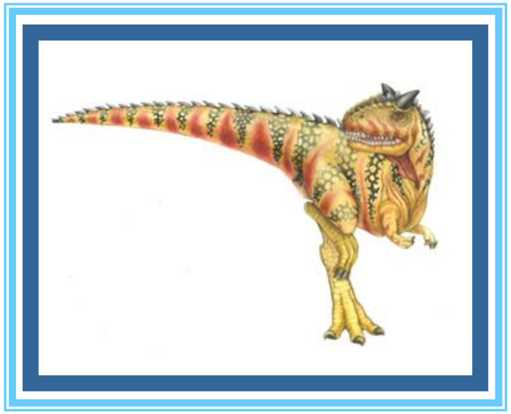
Chapter 16: Windows 7



Chapter 16: Windows 7

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

Objectives

- To explore the principles upon which Windows 7 is designed and the specific components involved in the system
- To understand how Windows 7 can run programs designed for other operating systems
- To provide a detailed explanation of the Windows 7 file system
- To illustrate the networking protocols supported in Windows 7
- To cover the interface available to system and application programmers

Windows 7

- 32-bit/64-bit preemptive multitasking operating system for Intel and AMD microprocessors
- Key goals for the system:
  - security
  - reliability
  - extensibility
  - portability
  - international support
  - energy efficiency
  - dynamic device support.
- Supports multiple OS personalities using user-mode subsystems.
- Windows 7 is for desktops. Windows Server 2008 R2 uses the same internals as 64-bit Windows 7, but with added features for servers.

History

- In 1988, Microsoft decided to develop a “new technology” (NT) portable operating system that supported both the OS/2 and POSIX APIs. NT supported servers as well as desktop workstations.
- 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 the Windows 3.0 Win16 API.
- Windows XP was released in 2001 to replace the earlier versions of Windows based on MS/DOS, such as Windows98.
- Windows XP was updated in 2005 to provide support AMD64 compatible CPUs, bringing support for 64-bit desktop systems.
- Windows Vista was released in late 2006, but was poorly received due to initial problems with application and device compatibility and sluggishness on the explosion of low-end “netbook” devices.
- Windows 7 was released in late 2009, greatly improving on Vista.

Design Principles

- Extensibility — layered architecture
  - Kernel layer runs in protected mode and provides access to the CPU by supporting threads, interrupts, and traps.
  - Executive runs in protected mode above the Kernel layer and, provides the basic system services
  - On top of the executive, environmental subsystems operate in user mode providing different OS APIs (as with Mach)
  - Modular structure allows additional environmental subsystems to be added without affecting the executive
- Portability —Windows 7 can be moved from on hardware platform to another with relatively few changes
  - Written in C and C++
  - Platform-dependent code is isolated in a dynamic link library (DLL) called the “hardware abstraction layer” (HAL)

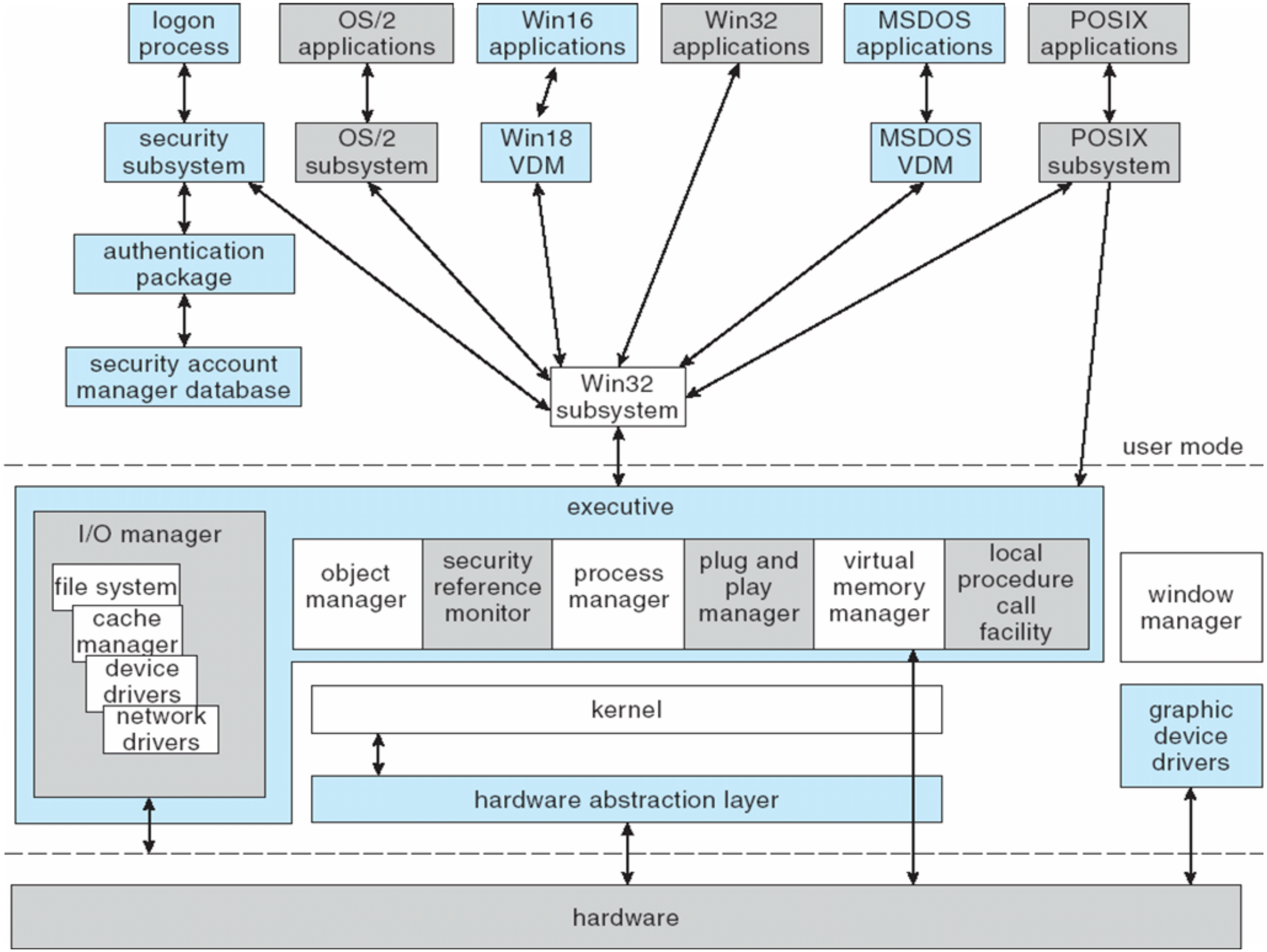
Design Principles (Cont.)

- Reliability —Windows 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 Windows without changing the source code. Applications created for previous versions of Windows run using various virtual machine techniques
- Performance —Windows 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, scaling to 100s of cores
- International support — supports different locales via the national language support (NLS) API, use of UNICODE throughout, and providing facilities for differences in date formats, currency, etc.

Windows Architecture

- Layered system of modules
- Protected mode — **hardware abstraction layer** (**HAL**), kernel, executive.
  - Executive includes file systems, network stack, and device drivers.
- User mode — collection of subsystems, services, DLLs, and the GUI
  - Environmental subsystems emulate different operating systems
  - Protection subsystems provide security functions
  - Windows services provide facilities for networking, device interfaces, background execution, and extension of the system
  - Rich shared libraries with thousands of APIs are implemented using DLLs to allow code sharing and simplify updates
  - A graphical user interface is built into Win32 and used by most programs that interact directly with the user

Depiction of Windows 7 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, 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.*

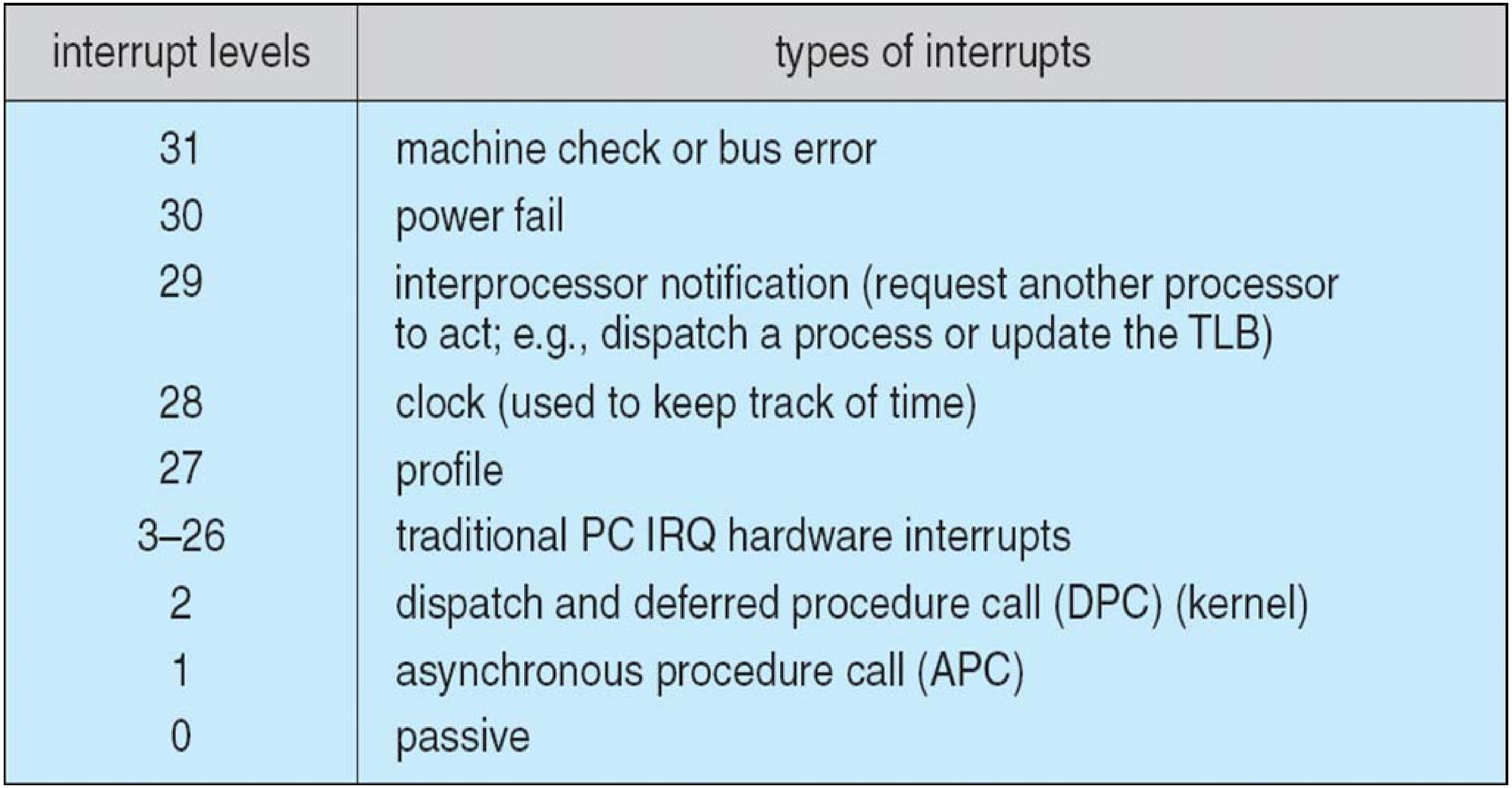
Kernel — Scheduling

- 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
    - Gives 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
    - Compute-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 Windows 7 does not guarantee that a real-time thread will start to execute within any particular time limit.
  - This is known as *soft real-time.*

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 manager supervises the use of all the objects
  - Generates an object *handle* used by applications to refer to objects
  - 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 executive allows any object to be given a name, which may be either permanent or temporary.
- Object names are structured like file path names in UNIX.
- Windows implements a *symbolic link object*, which is similar to *symbolic links* in UNIX that allow multiple nicknames or aliases to refer to the same object.
- 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 its parent process.
- Each object is protected by an access control list.
- The executive name space is extensible to allow naming of files, registry keys, and other objects with their own special semantics.

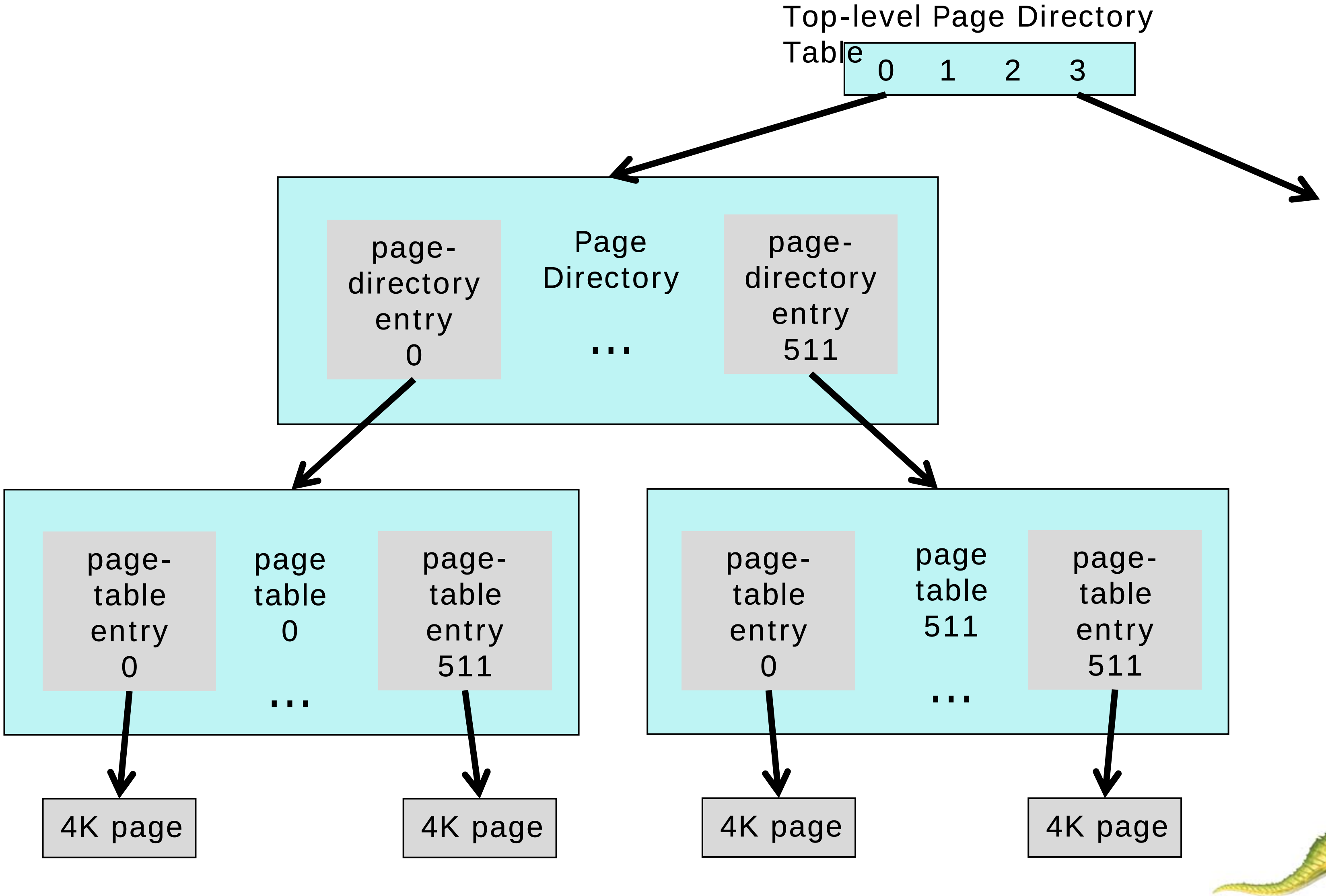
Executive — Virtual Memory Manager

- 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 address aliasing.

- The VM manager in Windows uses a page-based management scheme with a page size of 4 KB for both x86 and AMD64.

- The 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 physical memory or in the paging file on disk

Virtual-Memory Layout (32-bit)



Top-level Page Directory Table

0 1 2 3

page-directory entry 0

Page Directory

...

page-directory entry 511

page-table entry 0

page table 0

...

page-table entry 511

page-table entry 0

page table 511

...

page-table entry 511

4K page

4K page

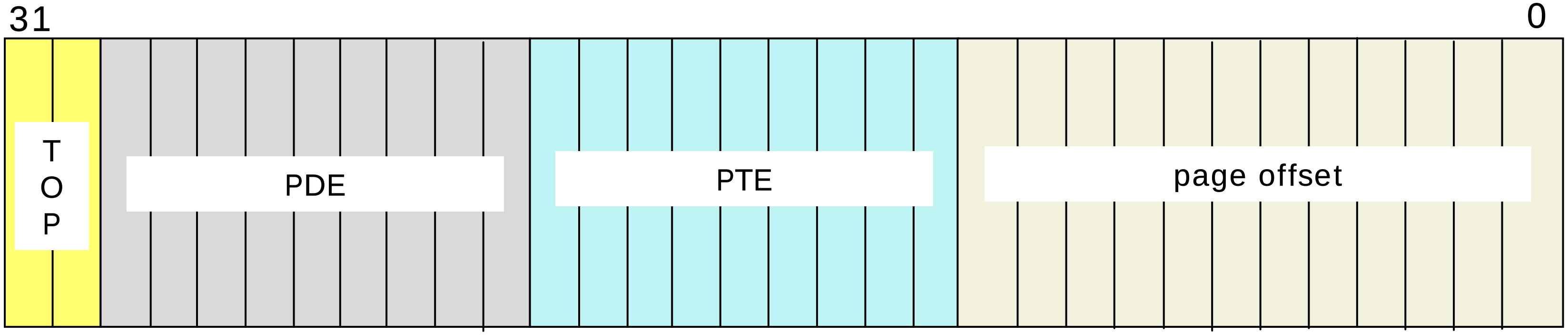
4K page

4K page

Virtual Memory Manager (Cont.)

- The virtual address translation in Windows uses several data structures within each process
  - A *top-level page directory* containing 4 *page directory entries* (PDEs) of size 8 bytes that may each point to a *page directory*.
  - Each page directory contains 512 page directory entries, that may each point to a *page table*.
  - Each page table contains 512 *page table entries* (PTEs) of size 8 bytes.
  - Each valid PTE points to a 4 KB *page frame* in physical memory.
    - Invalid PTEs are used by the OS to find pages on disk
- A 9-bit integer can represent all the values form 0 to 511, 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 byte address in physical memory.
- A physical page can be in one of six states: valid, zeroed, free, standby, modified and bad.

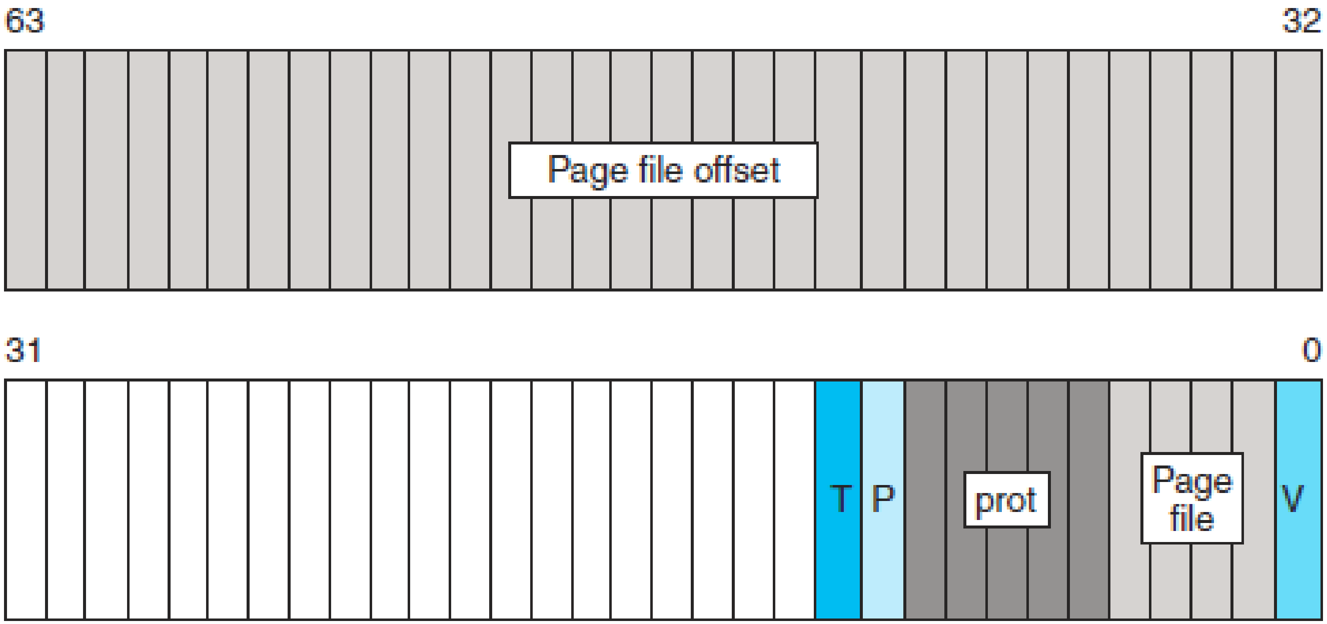
Virtual-to-Physical Address Translation



Translation for a 32-bit Virtual Address to a Physical Address

- 2 bit index into top-level page directory to get page directory
- 9 bit index into page directory to get page directory entry for page table
- 9 bit index into page table to get page table entry for physical page
- 12 bits for byte offset within physical page

Page File Page-Table Entry



PTE describing a page in the page file rather than memory (V = 0)

32 bit offset into page file, 4 bits to select a paging file,

5 bits for page protection, and page state bits:

T: page is in Transition from disk.

P: a *Prototype* PTE for shared pages.

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.

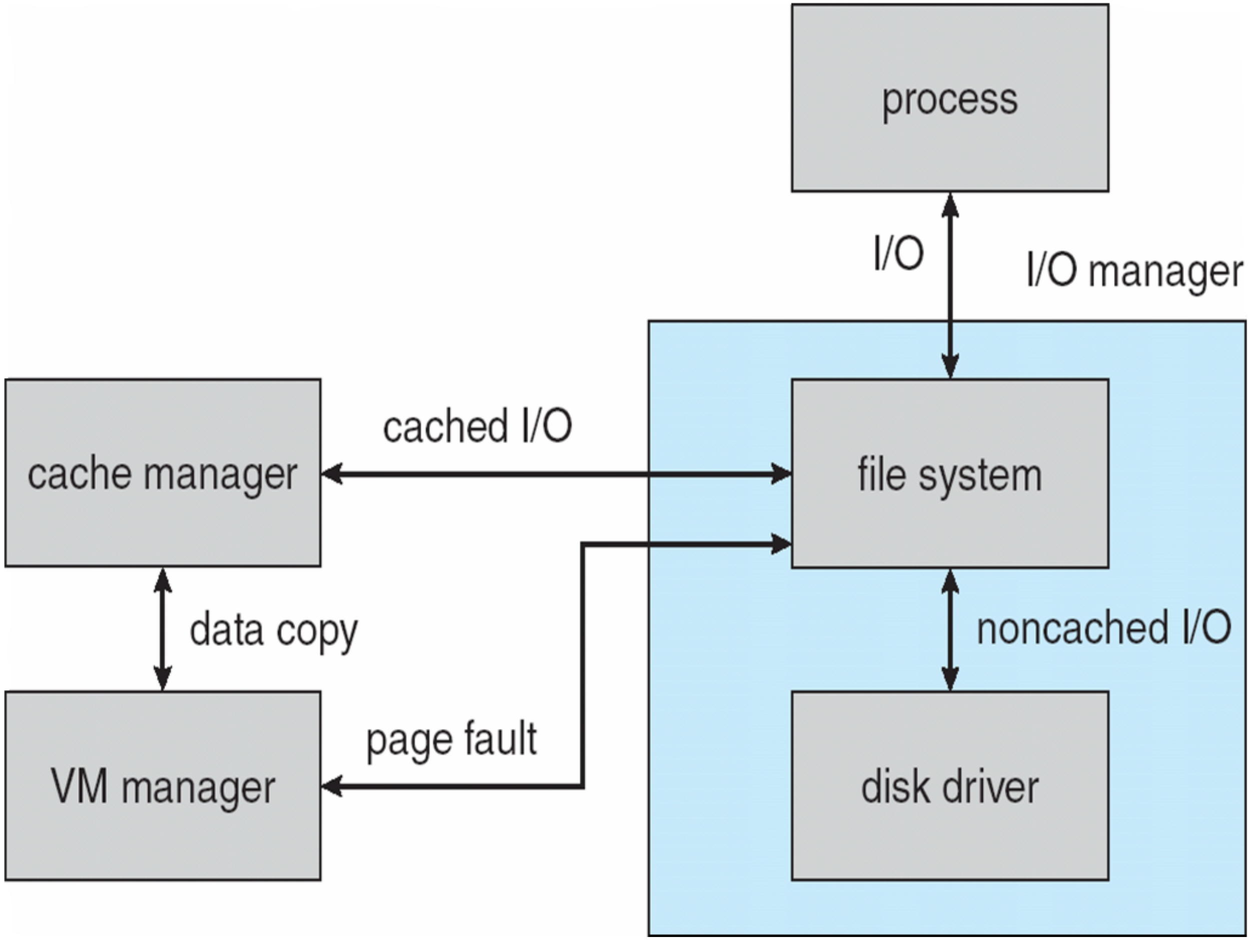
Executive — Local Procedure Call Facility

- The ALPC (Advanced Local Procedure Call) component passes requests and results between client and server processes within a single machine.
  - ALPC is used to request operations between the various Windows subsystems and services, as well as to provide the lower layer for standard RPC (Remote Procedure Calls) for a single machine.
  - Standard RPC can connect multiple machines, using TCP/IP or named pipes.
- When an ALPC channel is created, one of three types of message passing techniques must be specified.
  - First type is used for small messages; the port's message queue provides intermediate storage to copy between processes.
  - Second type avoids copying large messages by pointing to a shared memory section object created for the channel.
  - Third method reads and writes directly into each processes's address space, and is used by the Win32 GUI.

Executive — I/O Manager

- The I/O manager is responsible for
  - file systems
  - cache management
  - device and 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.
- Interfaces with the Windows cache manager, which handles caching for the entire I/O system.
- Implements dataflow model passing I/O requests between the drivers that implement the stack of software which operates each device.
- Supports synchronous and asynchronous operations, and timers, HW and SW interrupts, DMA, and other such facilities for drivers.

File I/O


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

Executive — Security Reference Monitor

- The object-oriented nature of the Windows kernel 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 – PnP and Power Managers

- PnP (Plug-and-Play) 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.
  - PnP also keeps track of the resources used by each device.
- The power manager controls energy use of by the CPU and devices.
  - Drivers for devices not being used are told to shut off device
  - CPUs are run at lower clock rate and/or lower energy states
  - System can be put into *standby* mode with only memory on, or
  - *Hibernated* by writing the contents of memory to disk and turning the system completely off

Environmental Subsystems

- User-mode processes layered over the native Windows executive services to enable Windows 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.

- The POSIX subsystem is designed to run POSIX applications following the POSIX.1 standard which is based on the UNIX model.

Environmental Subsystems (Cont.)

- 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 from a remote 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 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 older 16-bit FAT file system, the amount of internal fragmentation is reduced

File System — Internal Layout

- NTFS uses logical cluster numbers (LCNs) as disk addresses
- A file in NTFS is not a simple byte stream, as in MS-DOS or UNIX, rather, it is a structured object consisting of attributes
- Every file in NTFS is described by one or more records in an array stored in a special file called the Master File Table (MFT)
- Each file on an NTFS volume has a unique ID called a file reference.
  - 64-bit quantity that consists of a 48-bit file number and a 16-bit sequence number
  - Can be used to perform internal consistency checks
- The NTFS name space is organized by a hierarchy of directories; the index root contains the top level of the B+ tree

File System — Recovery

- All file system data structure updates are performed inside transactions that are logged.
  - Before a data structure is altered, the transaction writes a log record that contains redo and undo information.
  - After the data structure has been changed, a commit record is written to the log to signify that the transaction succeeded.
  - After a crash, the file system data structures can be restored to a consistent state by processing the log records.

File System — Recovery (Cont.)

- This scheme does not guarantee that all the user file data can be recovered after a crash, just that the file system data structures (the metadata files) are undamaged and reflect some consistent state prior to the crash.

- The log is stored in the third metadata file at the beginning of the volume.

- The logging functionality is provided by the Windows *log file service.*

File System — Security

- Security of an NTFS volume is derived from the Windows object model.
- Each file object has a security descriptor attribute stored in the MFT record.
- This attribute contains the security ID of the owner of the file, and an access control list that states the access privileges that are granted to each user and group that has access to the file.

Volume Management and Fault Tolerance

`FtDisk`, the fault tolerant disk driver for Windows, provides several ways to combine multiple disk drives into one logical volume

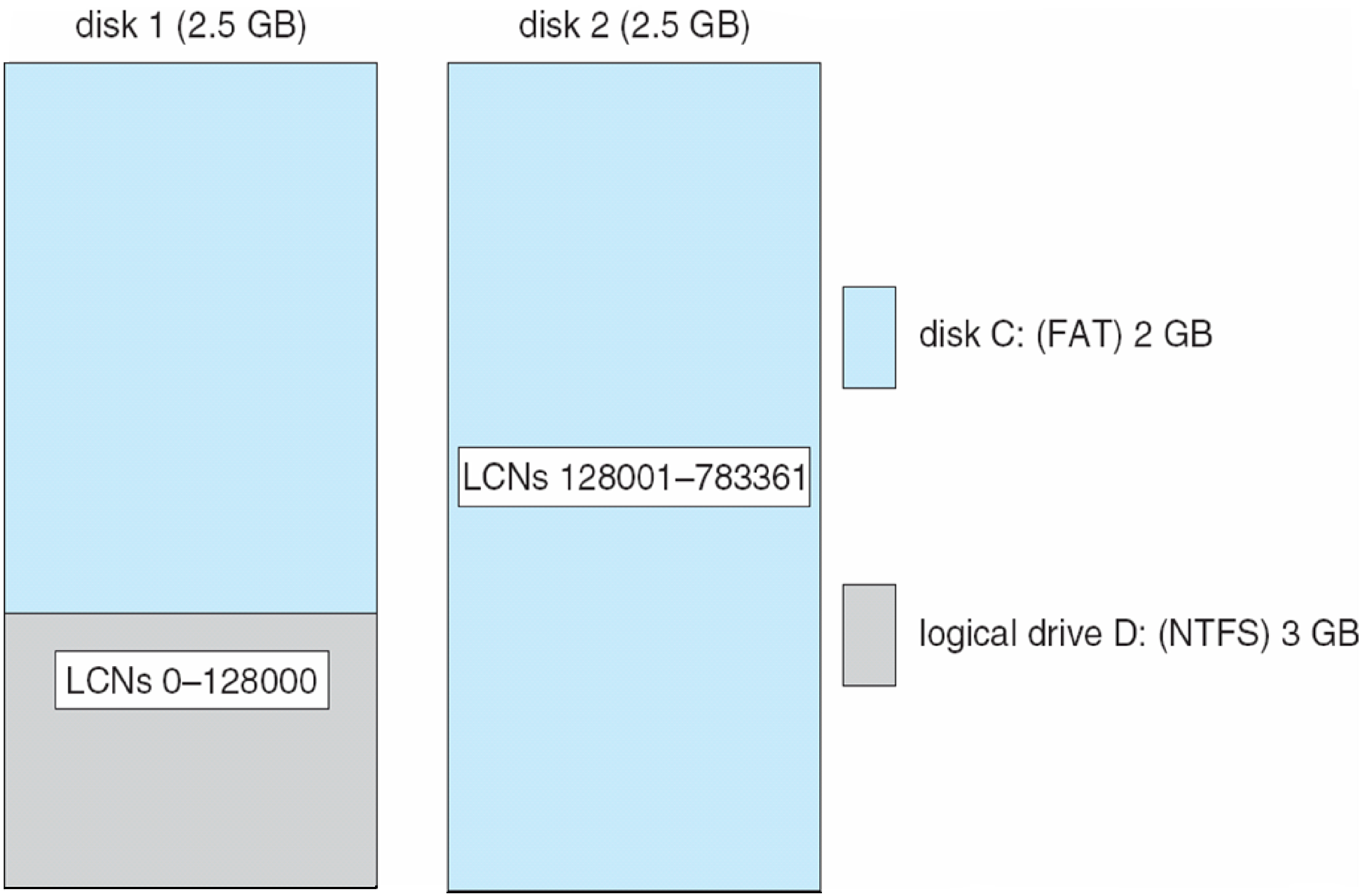
- Logically concatenate multiple disks to form a large logical volume, a *volume set*

- Interleave multiple physical partitions in round-robin fashion to form a *stripe set* (also called RAID level 0, or “disk striping”)
  - Variation: *stripe set with parity,* or RAID level 5

- Disk mirroring, or RAID level 1, is a robust scheme that uses a *mirror set* — two equally sized partitions on tow disks with identical data contents

- To deal with disk sectors that go bad, `FtDisk`, uses a hardware technique called *sector sparing* and NTFS uses a software technique called *cluster remapping*

Volume Set On Two Drives



disk 1 (2.5 GB)

disk 2 (2.5 GB)

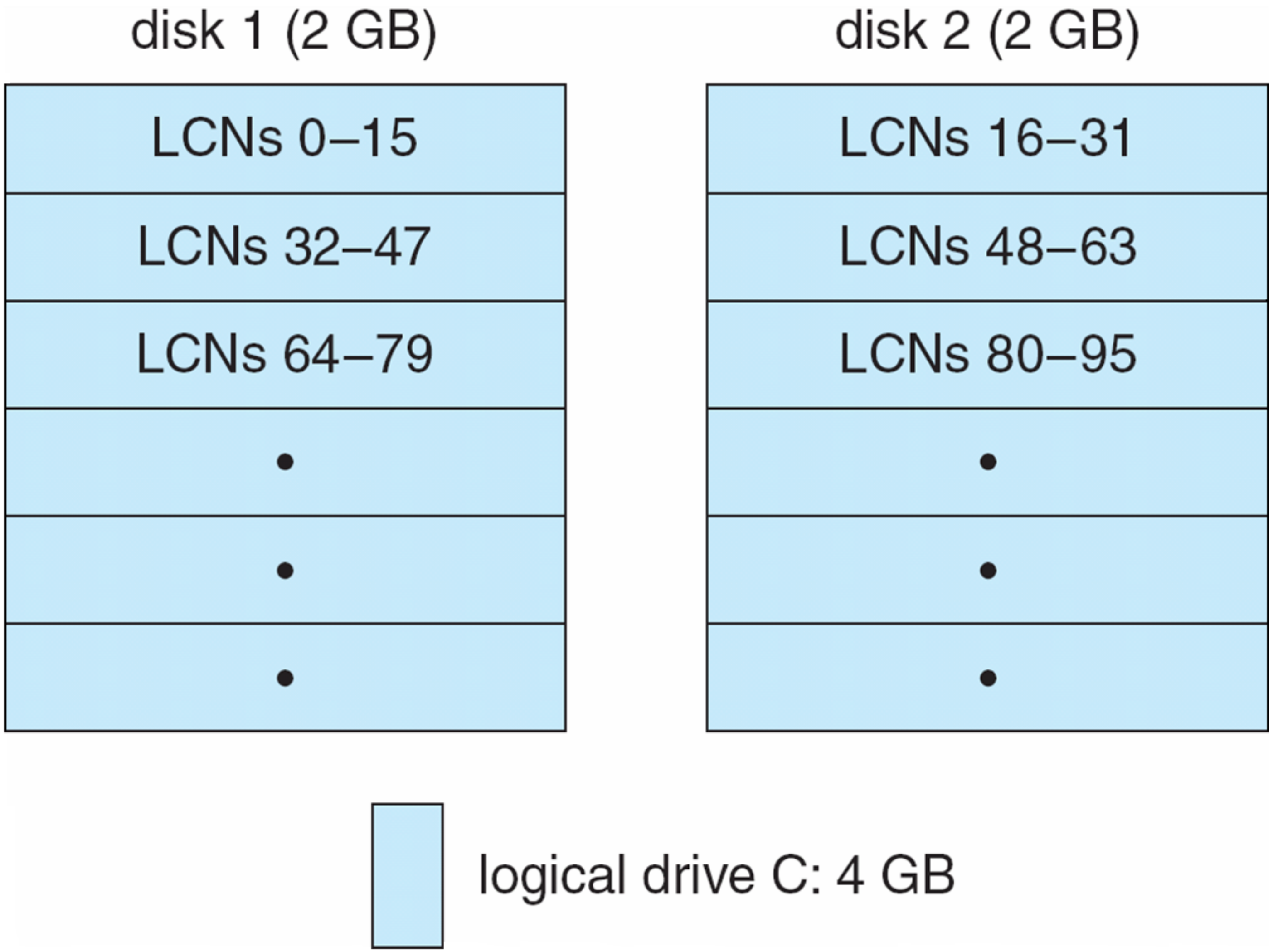
LCNs 0–128000

LCNs 128001–783361

disk C: (FAT) 2 GB

logical drive D: (NTFS) 3 GB

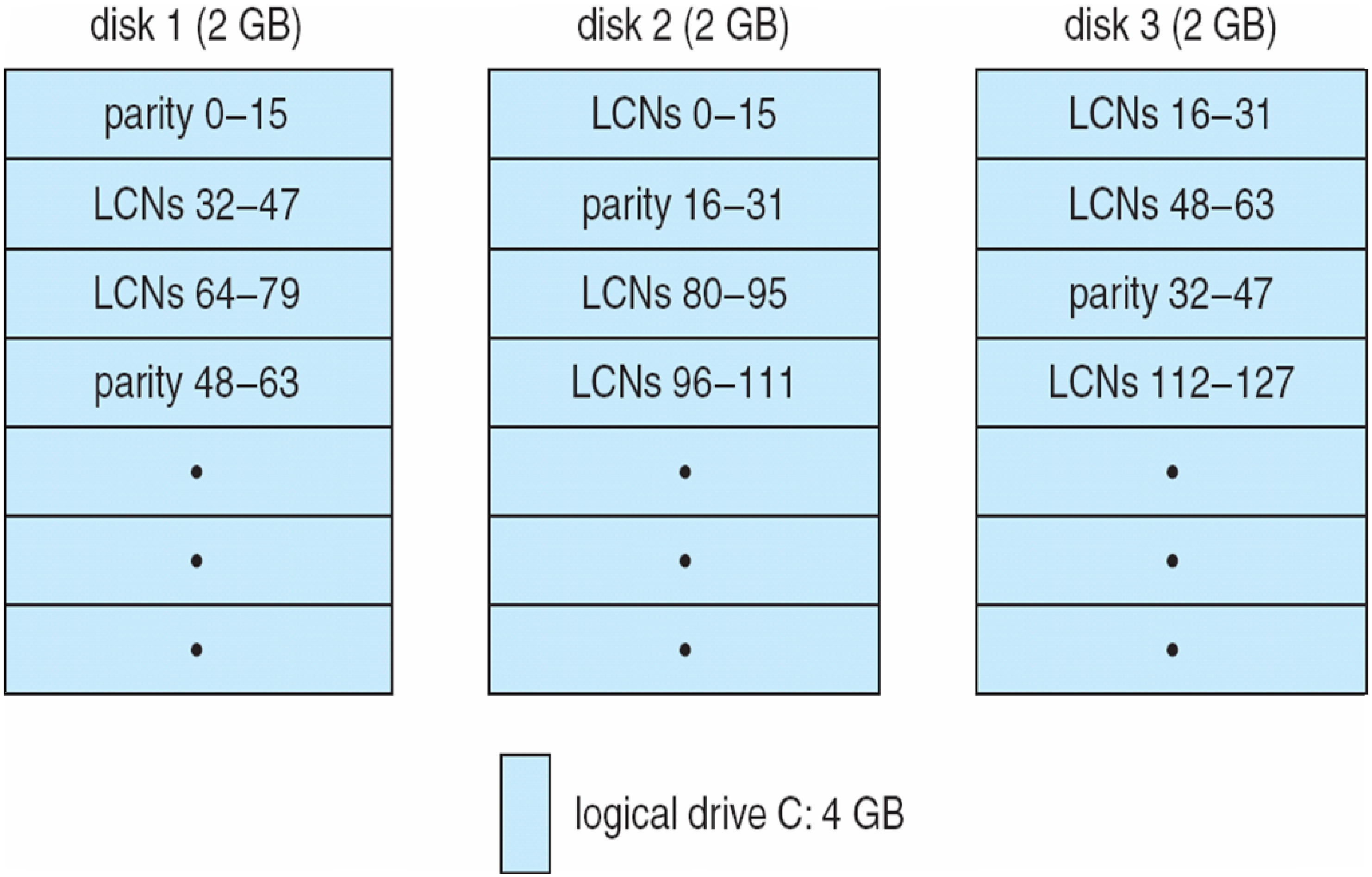
Stripe Set on Two Drives



disk 1 (2 GB)	disk 2 (2 GB)
LCNs 0–15	LCNs 16–31
LCNs 32–47	LCNs 48–63
LCNs 64–79	LCNs 80–95
•	•
•	•
•	•

□ logical drive C: 4 GB

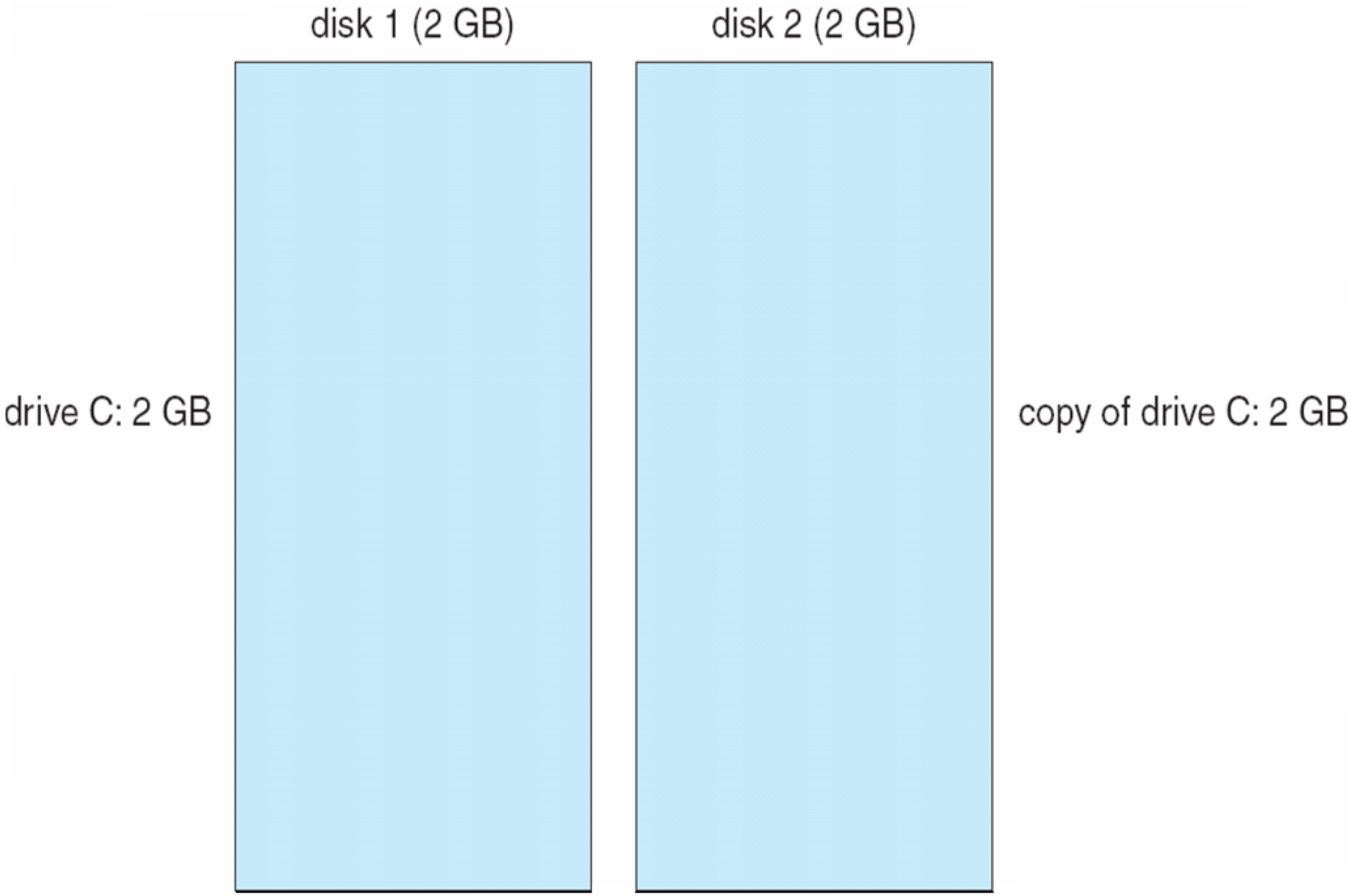
Stripe Set With Parity on Three Drives



disk 1 (2 GB)	disk 2 (2 GB)	disk 3 (2 GB)
parity 0–15	LCNs 0–15	LCNs 16–31
LCNs 32–47	parity 16–31	LCNs 48–63
LCNs 64–79	LCNs 80–95	parity 32–47
parity 48–63	LCNs 96–111	LCNs 112–127
•	•	•
•	•	•
•	•	•

▭ logical drive C: 4 GB

Mirror Set on Two Drives



File System — Compression

- To compress a file, NTFS divides the file's data into *compression units*, which are blocks of 16 contiguous clusters.
- For sparse files, NTFS uses another technique to save space.
  - Clusters that contain all zeros are not actually allocated or stored on disk.
  - Instead, gaps are left in the sequence of virtual cluster numbers stored in the MFT entry for the file.
  - When reading a file, if a gap in the virtual cluster numbers is found, NTFS just zero-fills that portion of the caller's buffer.

File System — Encryption

- NTFS provides per-file encryption services for encrypting individual files or directories of files using EFS (Encrypted File System).
- Windows will also encrypt entire volumes with BitLocker
  - Essentially all of the volume is encrypted
  - There are three levels of key protection
    - Hardware TPM
    - An electronic key plugged into a USB connection
    - User password
  - BitLocker machines should be shutdown rather than placed in standy to avoid attacks on the unencrypted physical memory.
- Systems protected by BitLocker have a high-degree of security against data theft of lost laptops or stolen systems.

File System — Reparse Points

- By default, a reparse point returns an error code when accessed.
- The error code tells the I/O manager that it needs to do something special to reference the file whose path contains the reparse point.
- Reparse points are be used to provide the functionality of UNIX *mounts* and *symbolic links* within the file system*.*
- Reparse points can also be used to access files that have been moved to offline storage, or to mark individual files for special processing by drivers written to extend the file system.

Networking

- Windows supports both peer-to-peer and client/server networking; it also has facilities for network management.

- To describe networking in Windows, 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 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:
  - Session control
  - File
  - Printer
  - 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 *SMB*

Networking — Protocols (Cont.)

- NetBEUI (NetBIOS Extended User Interface): default protocol for Windows 95 peer networking and Windows for Workgroups; used when Windows wants to share resources with these older networks.

- Windows uses the TCP/IP Internet protocol to connect to a wide variety of operating systems and hardware platforms, including wireless networks such as WiFi or 3G.

- PPTP (Point-to-Point Tunneling Protocol) is used to communicate between Remote Access Server modules running on Windows machines that are connected over the Internet.

Networking — Distributed Processing Mechanisms

- Windows 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 NetBEUI or TCP/IP.
- Named pipes are connection-oriented messaging mechanism that are named via the uniform naming convention (UNC) used to access remote files.
- 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 based on Berkeley UNIX, 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 RPC mechanism follows the widely-used Distributed Computing Environment standard for RPC messages, so programs written to use Windows RPCs are very portable.
  - RPC messages are sent using NetBIOS, or Winsock on TCP/IP networks, or named pipes on LAN Manager networks.
  - RPC messages can also be sent on top of higher level protocols, such as HTTP.
  - Windows provides the *Microsoft Interface Definition Language* to describe the remote procedure names, arguments, and results.
- Windows also provides DCOM (Distributed Component Object Model) support for invoking methods on object servers implemented on remote machines.

Networking — Redirectors and Servers

- In Windows 7, an application can use the Windows I/O API to access files from a remote computer as if they were local, provided that the remote computer is running an SMB 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.

Access to a Remote File

- The application calls the I/O manager to request that a file be opened (we assume that the file name is in the standard UNC format).
- The I/O manager builds an I/O request packet.
- The I/O manager recognizes that the access is for a remote file, and calls a driver called a Multiple Universal Naming Convention Provider (MUP).
- The MUP sends the I/O request packet asynchronously to all registered redirectors.
- A redirector that can satisfy the request responds to the MUP
  - To avoid asking all the redirectors the same question in the future, the MUP uses a cache to remember with redirector can handle this file.

Access to a Remote File (Cont.)

- The redirector sends the network request to the remote system.
- The remote system network drivers receive the request and pass it to the server driver.
- The server driver hands the request to the proper local file system driver.
- The proper device driver is called to access the data.
- The results are returned to the server driver, which sends the data back to the requesting redirector.

Networking — Domains

- NT uses the concept of a domain to manage global access rights within groups.
- A domain is a group of machines using NT Directory Server and sharing a common security policy and user database.
- Windows provides three models of setting up trust relationships
  - *One way, A trusts B*
  - *Two way, transitive, A trusts B, B trusts C so A, B, C trust each other*
  - *Crosslink – allows authentication to bypass hierarchy to cut down on authentication traffic.*

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 Objects

- 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 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.*

Thread Scheduling

- Scheduling in Windows utilizes four priority classes:
  - IDLE_PRIORITY_CLASS (priority level 4)
  - NORMAL_PRIORITY_CLASS (level 8 — typical for most processes
  - HIGH_PRIORITY_CLASS (level 13)
  - REALTIME_PRIORITY_CLASS (level 24)
- The process contains the default priority classes for its threads, but each thread can vary its own priority that is used for actual scheduling.
- To provide performance levels needed for interactive programs, Windows has a special scheduling rule for processes in the NORMAL_PRIORITY_CLASS
  - Windows 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 to foreground, Windows increases boosts it’s thread’s priorities and increments the scheduling quantum.

Thread Scheduling (Cont.)

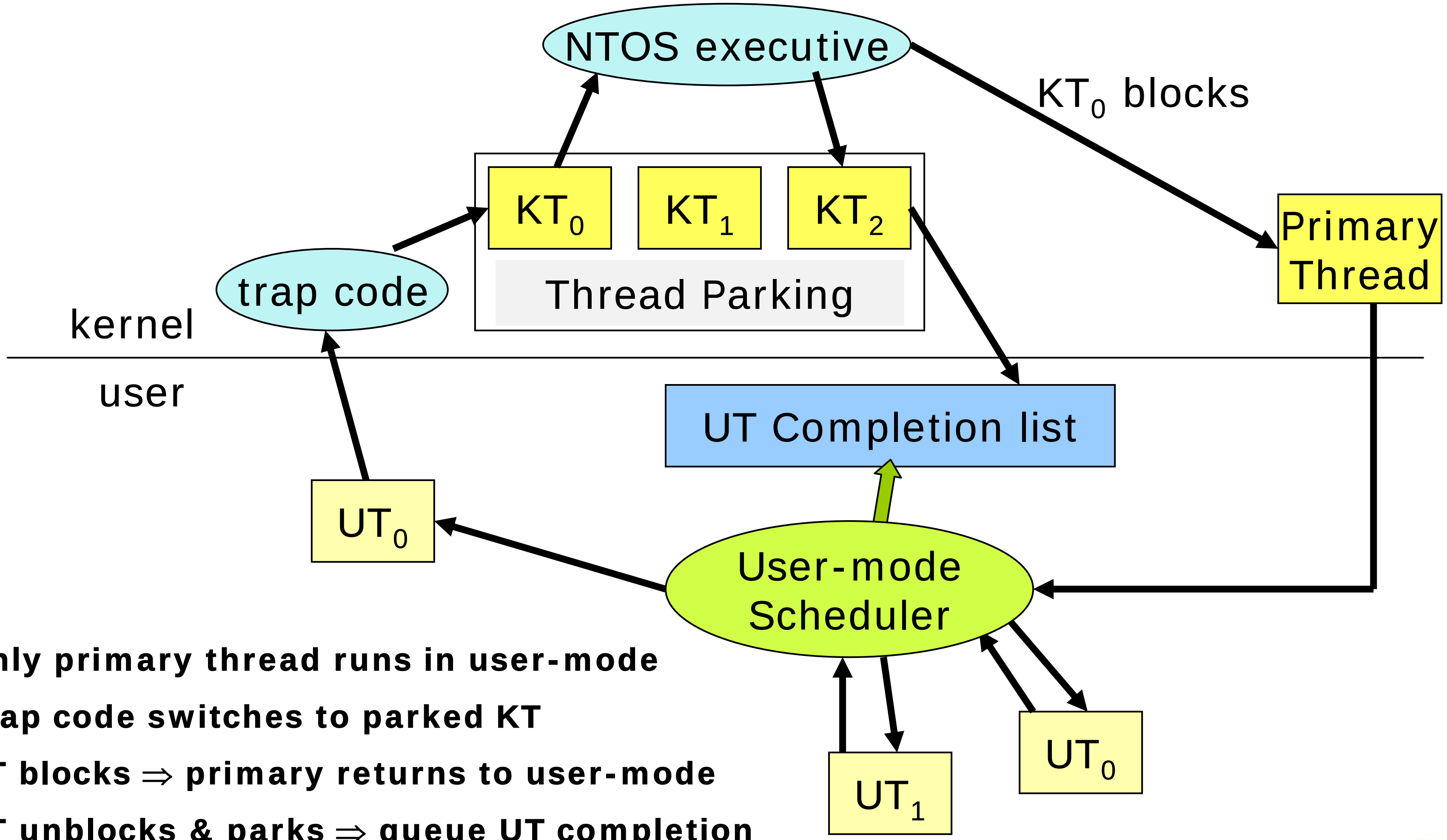
- The kernel dynamically adjusts the priority of a thread depending on whether it is I/O-bound or CPU-bound
  - I/O-bound threads have their priority boosted as long as they do not run to the end of their scheduling quantum.

- To synchronize the concurrent access to shared objects by threads, the kernel provides synchronization objects, such as semaphores and mutexes for synchronizing threads in different processes
  - In addition, threads can synchronize with kernel operations using the `WaitForSingleObject` or `WaitForMultipleObjects` APIs.
  - Synchronization between threads within a process uses Win32 APIs for critical sections, SRW (Simple Reader/Writer) Locks, Lockfree LIFO lists, and condition variables.

Thread Scheduling (Cont.)

- A fiber is user-mode code that gets scheduled according to a user-defined scheduling algorithm.
  - Windows includes fibers to facilitate the porting of legacy UNIX applications that are written for a fiber execution model.
  - Fibers are implemented purely in user mode by multiplexing the fibers onto true Windows threads.
- User Mode Scheduling was introduced into Windows 7 to provide support for the Visual Studio 2010 Concurrency RunTime (ConcRT)
  - UMS allows the user portion of a Windows thread to block in user-mode and the CPU to be switched to another user-mode thread without having to enter protected mode.
  - When the kernel portion of a Windows thread blocks in the OS, the CPU is returned to the user-mode scheduler for re-use.
  - When system calls complete, the user-mode scheduler is notified so it can again schedule the blocked thread for execution.

User Mode Scheduling (UMS)



Only primary thread runs in user-mode

Trap code switches to parked KT

KT blocks ⇒ primary returns to user-mode

KT unblocks & parks ⇒ queue UT completion

Programmer Interface — Interprocess Communication

- Win32 applications can have interprocess communication by sharing kernel objects, including shared memory sections and files.

- 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.

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 16