Windows OS

Koushik Viswanath S

Topics

- History
- Design Principles
- System components
- Environmental Subsystems
- File system
- Networking
- Process and Memory Managaement

History

- Microsoft decided in 1988 to develop a new technology(NT) portable OS the supported both OS/2 and POSIX APIs
- NT was supposed to use OS/2 API as native environment but later during development NT was changed to use Win32 API
- Development and enhancement of previous versions of windows systems lead to release of Windows 7 in 2009.

Features:

- 32bit pre-emptive multitasking os for intel
- Uses micro-kernel architecture
- Has 6 client and 3 server versions
- Key goals:
  - Portability
  - Security
  - POSIX compliance
  - Multiprocessor support
  - Extensibility
  - International support
  - Compatibile with other MS applications

Design Principles

- Extensibility- layered architecture, modular structure
- Portability- can be moved from one hardware architecture to another with few changes
- Reliability- Uses hardware protection for virtual memory and software protection mechanisms for OS resources
- Compatibility- applications that follow POSIX standards can be run without a change in the source code
- Performance- subsystems can participate with one another via high performance message passing
- International Support- supports different locales with national lang. support API(uses UNICODE characters)

Windows 7 architecture

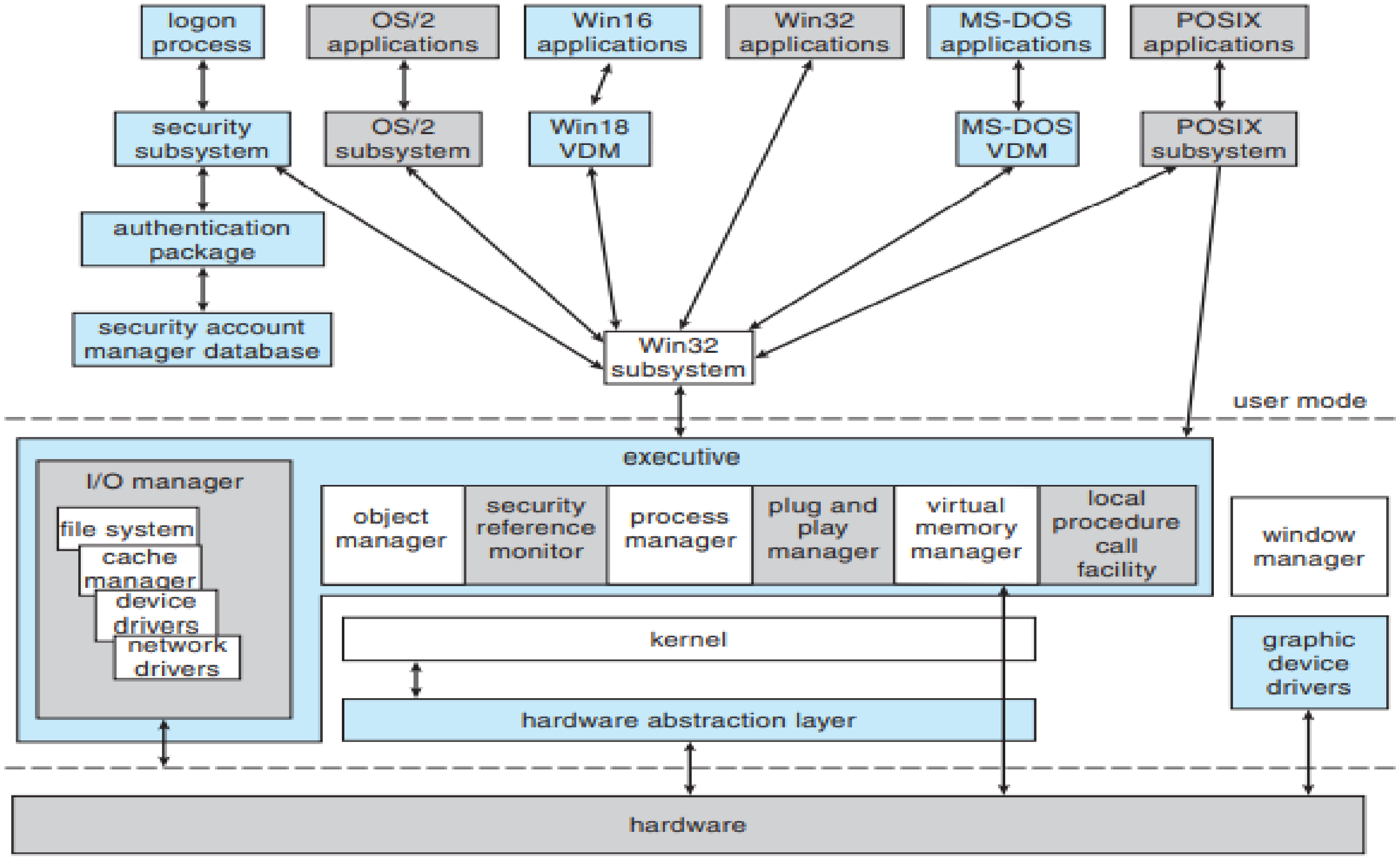

Figure 19.1 Windows block diagram.

System components- Kernel

- Foundation for the executive and the subsystems
- Object oriented(dispatcher objects, control objects)
- Never paged out of memory, execution never pre-empted
- Responsibilities:
  - Thread scheduling
  - Interrupt and exception handling
  - Low level process synchronization
  - Recovery after power failure

Kernel – Process and Threads

- The process has a virtual memory address space, information and an affinity to one or more processors
- Threads are the unit of execution scheduled by the kernel's dispatcher
- Each thread has its own state including priority, processor affinity and accounting information
- There are six states for a thread:

  ready, standby, running, waiting, transition, terminated

Kernel- scheduling

- Dispatcher uses 32level priority scheme to determine order of execution
  - Variable class- threads with priorities 0 to 15 (0 for mem mngmt)
  - Real time class- threads with priorities 16 to 31
- Characteristics of priority strategy
  - Very good response time
  - Enables i/o bound threads to keep them busy
  - Complete-bound threads soak up the spare CPU cycles in the bg.
- Scheduling can occur when thread enters ready or wait state or when a thread terminates or when it’s priority or affinity is changed
- Real-time threads are given preferential access to CPU(but no guarantee that it will strat executing in any particular time limit-soft real time

Interrupt request levels

- The kernel uses an interrupt-dispatch table to bind each interrupt level to a service routine. In a multiprocessor computer, Windows keeps a separate interrupt-dispatch table (IDT) for each processor, and each processor's IRQL can be set independently to mask out interrupts.
- All interrupts that occur at a level equal to or less than the IRQL of a processor are blocked until the IRQL is lowered by a kernel-level thread or by an ISR returning from interrupt processing.

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

Figure 19.2 Windows interrupt-request levels.

Trap handling

- Provides trap handler for exceptions and interrupts generated by hardware or software
- Exceptions that cannot be handled by trap handler are handled by exception dispatcher
- Kernel uses spin locks(hardware locks) that residue in global memory to attain multiprocessor mut-ex.
- Interrupt dispatcher handles interrupts by calling either interrupt service routine or internal kernel routine

Virtual memory

- Design of the VM manager assumes that the underlying hardware supports virtual to physical mapping
- It uses a page based management scheme with page size 4KB
- It uses two step process to allocate memory:
  - Reserves a portion of the process's address space
  - Commits the allocation by assigning space in system's paging file

- Virtual memory layout

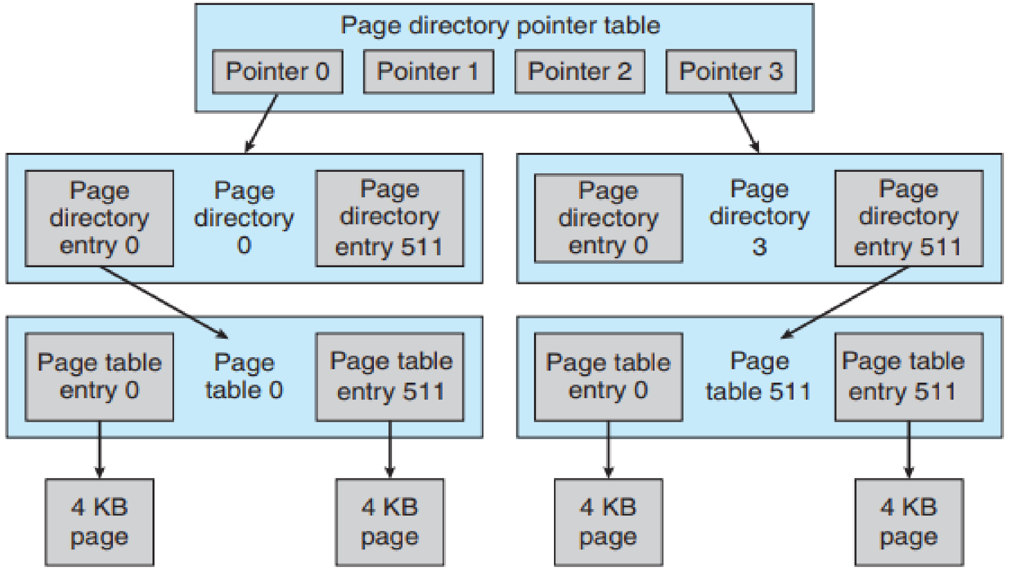

Figure 19.3 Page-table layout.

Virtual Memory manager

- Address translation involves several data structures:
  - Each process has a page directory that contains 1024 entries each of size 4B
  - Each page directory points to a page table which contains 1024 page table entries(PTE’s) of size 4B
  - Each PTE points to a 4KB page frame in physical memory
- A 10 bit integer can represent all values from 0 to 1023, therefore we can select any entry in a page directory or table
- This property is used in translation of virtual address pointer to a bye address in physical memory
- A page has six states:

  valid,free,zeroed,standby,modified,bad

Virtual to physical translation

- 2 bits for PTR
- 9 bits PDE index
- 9 bits PTE index
- 12 bits page offset

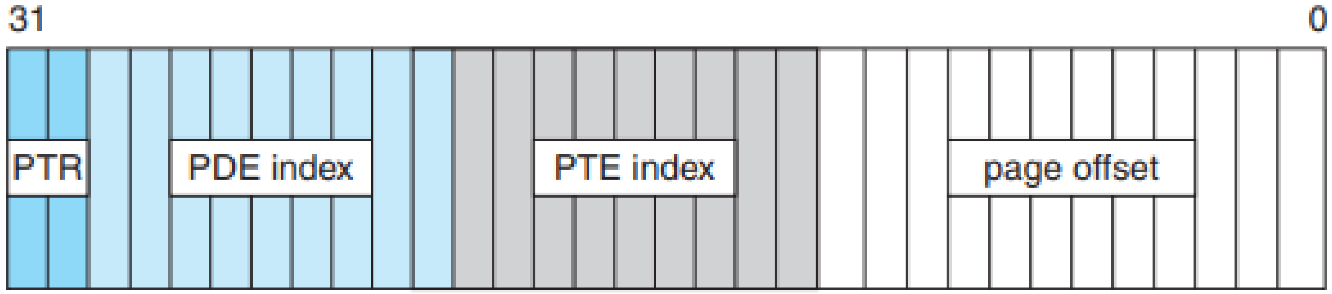

Figure 19.4 Virtual-to-physical address translation on IA-32.

I/O manager

- Keeps track of installable flie systems, manages buffers for I/O requests
- Works with VM manager to provide memory mapped file I/O
- Controls cache manager which handles caching for the entire I/O system
- Supports sync and async operations, provides break for drivers alternatively

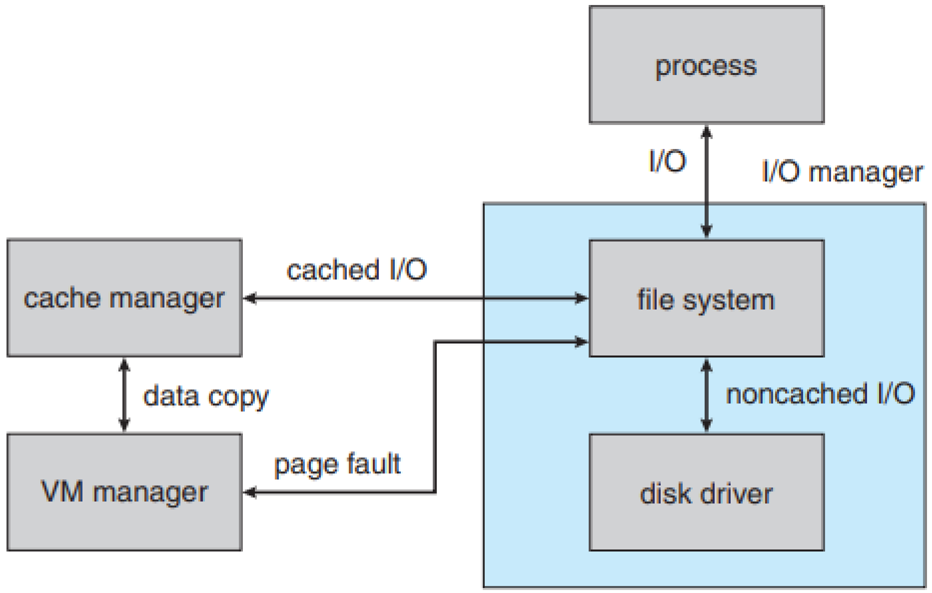

Figure 19.6 File I/O.

Environmental subsystems

- User mode processes layered over the native services which enables the program developed for other OS to run
- Win32 is used to start all processes. It also provides the hardware capabilities
- MS-DOS environment is provided by VDM(virtual dos machine) which is a process that is paged and dispatched similar to threads
- The POSIX subsystem is designed to run applications which follow POSIX standards

File systems

- Each file in the NTFS volumes has a unique ID called a file reference
- Each directory uses a B+ tree data structure to store files
- The log file records all metadata updates to the file system
- The root directory is the top level directory in the file-system hierarchy
- Bad cluster file keeps track of the bad areas in the volume which helps the NTFS for error recovery

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
  - This scheme does not guarantee that all the user file data can be recovered after a crash
  - The log is stored in the third metadata file at the beginning of volume
  - The logging functionality is provided by the Windows 7 log file service

File system - Security

- Each file has a security descriptor attribute stored in record
- This attribute contains the access token of the owner of the file, and an access control list that states the access privileges that are granted to each user that has access to the file

Volume management and fault tolerance

- Fault tolerance disk driver for Windows 7 provides several ways to combine multiple SCSI 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
- Disk mirroring 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(new technology file system) uses a technique called cluster remapping

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
  - Clusters that contain all zeroes are not actually allocated or stored on disk
  - Instead, gaps are left in the sequence of virtual cluster numbers stored in the Managed File Transfer(MFT) entry of the file
  - When reading a file, if a gap is found NTFS zerofills that portion

Networking

- Windows 7 supports both peer to peer and client server networking
- To define networking in Windows 7 we refer two internal networking interfaces:
  - NDIS(Network Device Interface Specification)- seperates the network adapter from the transport protocols so that either can be changed
  - TDI(Transport Driver Interface)- Enables any session layer component to use any available transport mechanism

Networking Protocols

- Server message Block protocol is used to send I/O requests over the network. It has four message types
  - Session control
  - File
  - Printer
  - Message
- Network basic I/O is a hardware abstraction interface for networks
  - It establishes logical names in the network
  - It establishes logical connection between two networks
  - It supports reliable data transfer

Networking domains

- It manages global access rights within groups
- A domain is a group of machines running NT server that share common policies
- Windows provide three models
  - One way (a trusts b)
  - Two way (transitive)
  - Crosslink

Process Management

- Kernel dynamically adjusts the priority of a thread depending on whether it is I/O bound or CPU bound
- Win32 uses 4 priority classes:
  - Idle priority
  - Normal priority
  - High priority
  - Realtime priority
- Kernel provides synchronization objects such as semaphores and mutex. The Win32 also has critical section to enhance synchronization

Interprocess communication

- IPC occurs through kernel objects
- Every thread of Win32 has its own input queue from which it receives message
- This is more reliable than the shared input queue of 16 bit windows as it avoids queues getting stuck

Memory management

- VirtualAlloc reserves or commits virtual memory
- VirtualFree decommits or releases the memory
- Multistage management
- The thread local storage mechanism allocates global storage on per-thread basis. It provides both dynamic and static methods of creating thread-local storage

Summary

Discussed about :

- Design Principles
- System components
- Environmental Subsystems
- File system
- Networking
- Process and Memory Managaement