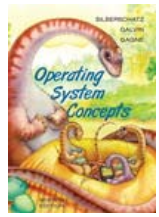
Appendix C: Introduction to *Windows 2000* *(Case Study)*

- History (C.1)
- Design Principles (C.2)
- System Components (C.3)

File System, Environmental Subsystems and Programmer Interface are not covered



Windows 2000

- The successor to the Windows NT, it was named Windows NT vers.5.0
- 32-bit preemptive multitasking operating system for Intel-Pentium microprocessors.
- Key goals for the system:
  - Portability (only among Intel architectures)
  - security
  - POSIX compliance (Portable OS Interface based on a UNIX model)
  - multiprocessor support
  - extensibility
  - international support
  - compatibility with MS-DOS and MS-Windows applications.
- It uses a micro-kernel architecture.
- Available in four versions, Professional (desktop), Server, Advanced Server, National Server.
- In 1996, more NT server licenses were sold than UNIX licenses

History (C.1)

- In 1988, Microsoft decided to develop a “new technology” (NT) portable operating system that supported both the OS/2 and POSIX (a standardized UNIX system call interface) APIs (Application-Programming Interfaces).
- Dave Cutler was the architect for Windows 2000
- 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 (system call interface containing more than 2000 functions ranging from those used to create a process to others that query a performance counter), reflecting the popularity of Windows 3.0.
- Different versions of Windows operating systems implement different subsets of the Win 32 API.

Design Principles (C.2)

- Extensibility — layered architecture (to keep up with advances in computing technology).
  - Windows 2000 Executive, which runs in kernel 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. (these subsystems emulate different operating systems, therefore programs written for MS-DOS, Microsoft Windows, and POSIX can all run on Win2000 in the appropriate environment.

Design Principles (C.2) (Cont.)

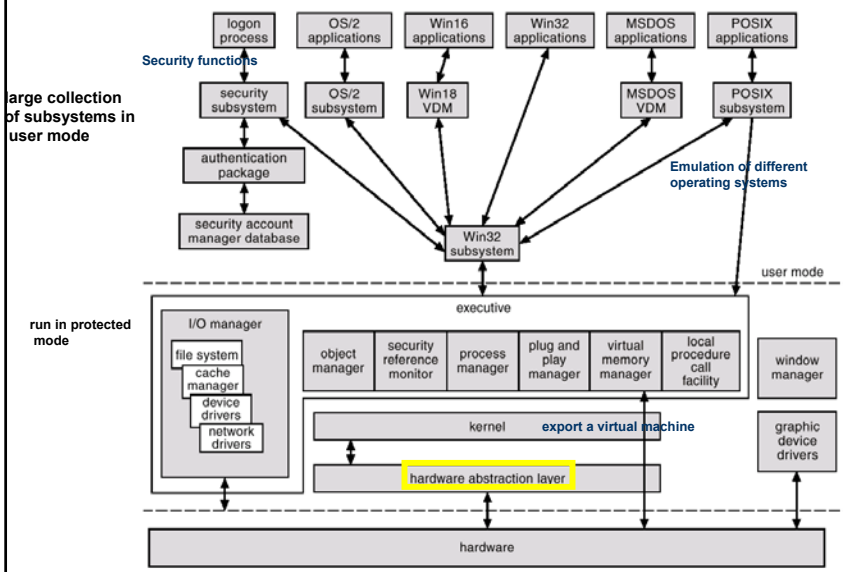
- Portability — 2000 can be moved from on hardware architecture (among Intel) to another with relatively few changes.
  - Written in C and C++.
  - Processor-dependent code is isolated in a dynamic link library (DLL) called the "hardware abstraction layer" (HAL). A Windows DLL is a file that contains collection of software code that perform common functions such as opening or saving file. The DLL gets mapped into a process's address space such that any functions in the DLL appears to be part of the process.
- Reliability — 2000 uses hardware protection for virtual memory, and software protection mechanisms for operating system resources.

Design Principles (C.2) (Cont.)

- Compatibility — applications that follow the IEEE 1003.1 (POSIX) standard can be compiled to run on 2000 without changing the source code-it can run the executable binaries for many programs compiled for Intel x86 architectures running MSDOS 16 bits windows and 32 bits windows (win16-win32) by using the 2000 sub-systems.
- Performance — 2000 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 Use — The National Language Support (NLS) provides routines to format dates times and money. String comparisons for various character sets is done through UNICODE (converts ANSI characters to UNICODE)

System Components (C.3)

- Layered system of modules.
  - Kernel mode — HAL, kernel, executive.
  - User mode — collection of subsystems
    - ❖Environmental subsystems emulate different operating systems.
    - ❖Protection subsystems provide security functions.



Depiction of 2000 Architecture

System Components (C.3) (Cont.)

- Kernel
  - Foundation for the executive and the subsystems.
  - The kernel is never paged out of memory; its execution is never preempted (always running).
  - Four main responsibilities:
    - thread scheduling
    - interrupt and exception handling
    - low-level processor synchronization
    - recovery after a power failure

System Components (C.3) (Cont.)

- 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.)

System Components (C.3) (Cont.)

- Kernel — Process and Threads
  - The process has a virtual (or logical) 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 (next to run), running, waiting (for I/O completion), transition(waiting for resources)*, and *terminated*.

System Components (C.3) (Cont.)

- Kernel — Scheduling
  - The dispatcher uses a 32-level priority scheme (0-31) to determine the order of thread execution. Priorities are divided into two classes (0-15 and 16-31).
    - 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 2000's priority strategy.
    - Tends 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.
    - When a variable-priority thread is released from a wait operation, the dispatcher boosts its priority. The amount of the boost depends on for what the thread was waiting (a thread that was waiting for a keyboard I/O would get a large priority increase than a thread that was waiting for a disk operation).

System Components (C.3) (Cont.)

- 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 Win2000 does not guarantee that a real-time thread will start to execute within any particular time limit. (This is known as *soft realtime*.)

For portability, the interrupt dispatcher maps any hardware interrupt (Intel, DEC Alpha) into a standard set. The interrupts are prioritized and serviced in priority order (8 kernel interrupts and 24 hardware interrupt via the HAL)

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

Windows 2000 Interrupt Request Levels

System Components (C.3) (Cont.)

- Executive (provide a set of services to subsystems)
  - Object Manager (supervise the use of all objects)
    - Win2000 uses objects for all its services and entities; the object manager supervises the use of all the objects.
    - Generates an object *handle (thread uses the object manager method* `open` *to get a handle to the object)*
    - Checks security.
    - Keeps track of which processes are using which object.
    - Objects are manipulated by a standard set of methods, namely `create, open, close, delete, query name, parse and security.`

System Components (C.3) (Cont.)

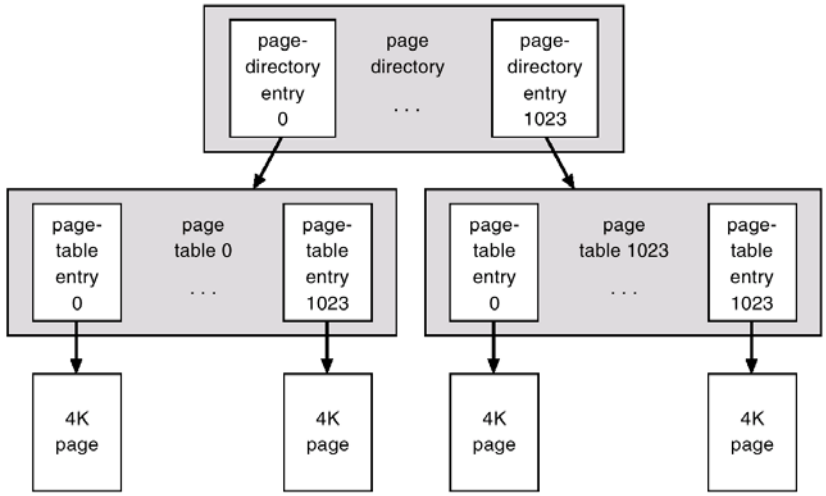
- Executive (Cont'd)
  - Naming Objects
    - The Win2000 executive allows any object to be given a name, which may be either permanent or temporary.
    - Object names are structured like file path names in MS-DOS and UNIX.
    - Win2000 implements a *symbolic link object*, which is similar to *symbolic 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.

System Components (C.3) (Cont.)

- Executive (Cont'd)
  - Virtual Memory Manager
    - The design of the VM manager assumes that the underlying hardware supports virtual to physical mapping, a paging mechanism, and transparent cache coherence on multiprocessor systems.
    - The VM manager in Win2000 uses a page-based management scheme with a page size of 4 KB.
    - Pages of data that are assigned to a process but are not in physical memory are stored in the paging file (on disk)
    - Win2000 can limit the amount of paging file space that a process consumes through a quota on committed memory

System Components (C.3) (Cont.)

- Virtual Memory Manager (Cont.)
  - The virtual address translation in Win2000 uses several data structures.
    - Each process has a *page directory* that contains 1024 *page directory entries* of size 4 Kbytes.
    - Each page directory entry points to a *page table* which contains 1024 *page table entries* (PTE's) of size 4 Kbytes.
    - Each PTE points to a 4 KB *page frame* in physical memory.



Virtual-Memory Layout

System Components (C.3) (Cont.)

- Executive (cont'd)
  - 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.

System Components (C.3) (Cont.)

- Executive (Cont'd)
  - 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.

System Components (C.3) (Cont.)

- Executive (Cont'd)
  - Security Reference Monitor
    - The object-oriented nature of Win2000 enables the use of a uniform mechanism to perform run-time 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.

System Components (C.3) (Cont.)

- Executive (Cont'd)
  - 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 (through a PCI bus that connects the processor-memory subsystem to the fast devices added) or USB, the PnP manager loads the appropriate driver.
    - The manager also keeps track of the resources used by each device.