Unit 1: Introduction and Overview

1.3. Windows 2000 – Concepts & Tools

Windows 2000 - Concepts & Tools

- Win32 API (application programming interface):
  - Common programming interface to Windows NT/2000/XP, Windows 95/98/ME and Windows CE
  - OS implement (different) subsets of the API
  - MSDN: www.microsoft.com/msdn
- Windows 2000 supports multiple subsystems (APIs):
  - Win32 (primary), POSIX, OS/2
  - User space application access OS functionality via subsystems
- OS/2 used to be primary subsystem for Windows NT

Services, Functions, and Routines

- Win32 API functions:
  - Documented, callable subroutinges
  - *CreateProcess, CreateFile, GetMessage*
- Windows 2000 system services:
  - Undocumented functions, callable from user space
  - *NtCreateProcess* is used by *CreateProcess* as an internal service
- Windows 2000 internal routines:
  - Subroutines inside the Windows 2000 executive, kernel, or HAL
  - Callable from kernel mode only (device driver, NT OS components)
  - *ExAllocatePool* allocates memory on Windows 2000 system heap

Services, Functions, and Routines (contd.)

- Windows 2000 services:
  - Processes which are started by the Service Control Manager
  - Example: The *Schedule* service supports the at-command
- DLL (dynamic link library)
  - Subroutines in binary format contained in dynamically loadable files
  - Examples: MSVCRT.DLL – MS Visual C++ run-time library
    KERNEL32.DLL – one of the Win32 API libraries

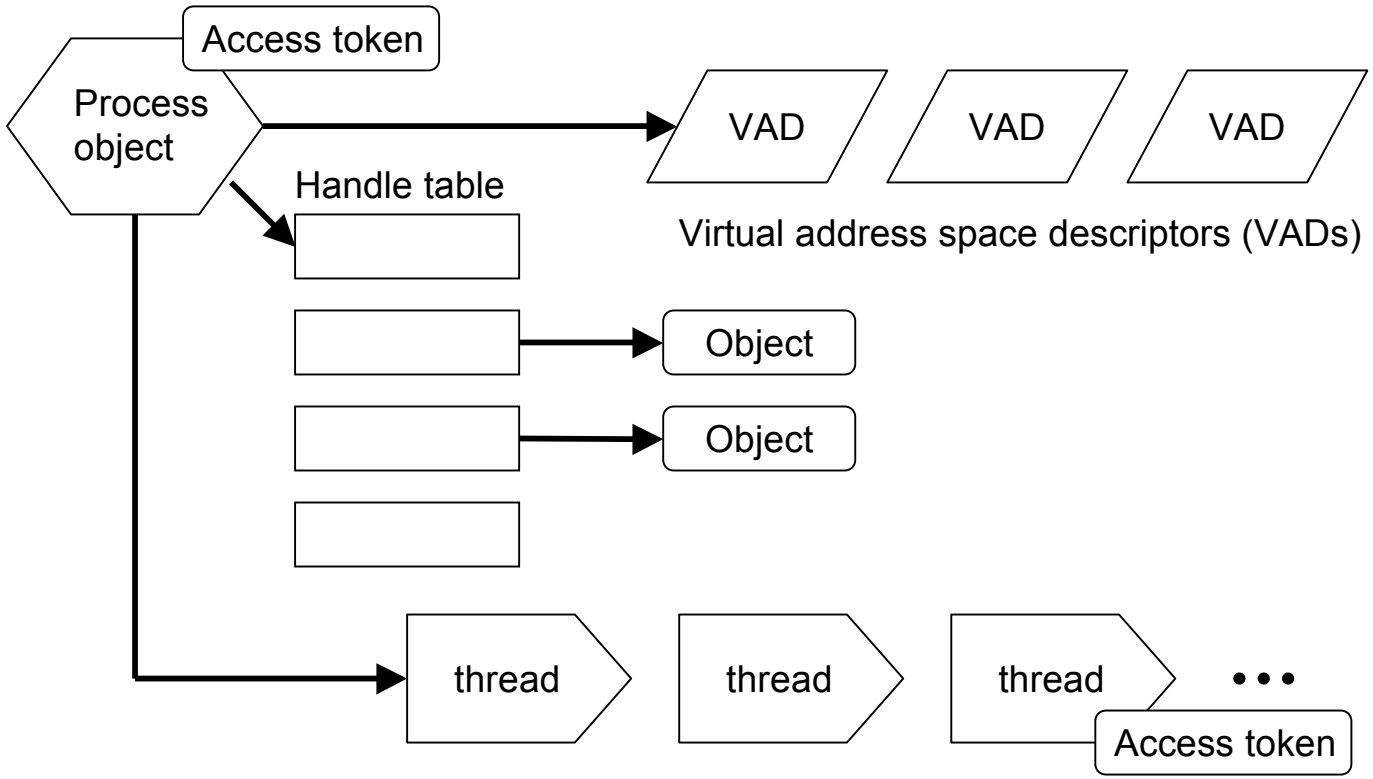
Processes and Threads

- *Program* – sequence of instructions
- *Process* – container for *threads* executing a program
- A Windows 2000 process is described by:
  - Executable program (code + data)
  - Private virtual address space
  - System resources (semaphores, communication ports, files)
  - Unique identifier – process ID (intern: client ID)
  - At least one thread
- Job (introduced with Windows 2000)
  - Collection of processes that share a set of quotas, limits, and security settings

Processes and Threads (contd.)

- Thread is the unit of *scheduling* in Windows 2000
  - Multiple threads mayshare the address space of a container process.
- A *thread* is described by:
  - Register content (processor state)
  - Two stacks (user mode/kernel mode)
  - Private memory address space used by
    - Subsystems,
    - Runtime library,
    - DLLs
  - Unique identifier – thread ID (internally: client ID)
    - Process Ids and thread Ids don‘t overlap
- Thread context is architecture-specific
  - See GetThreadContext() from Win32 API

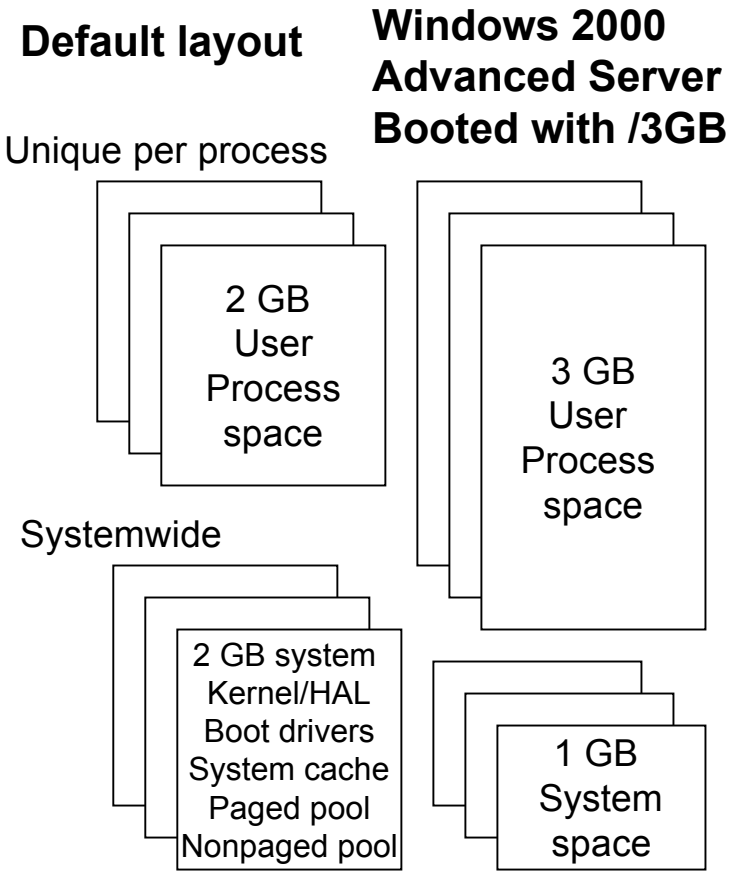
A Process and its Resources



Virtual Memory

- 32-bit address space (4 GB)
- 2 GB user space (per process)
- 2 GB operating system
- Memory manager maps virtual onto physical memory

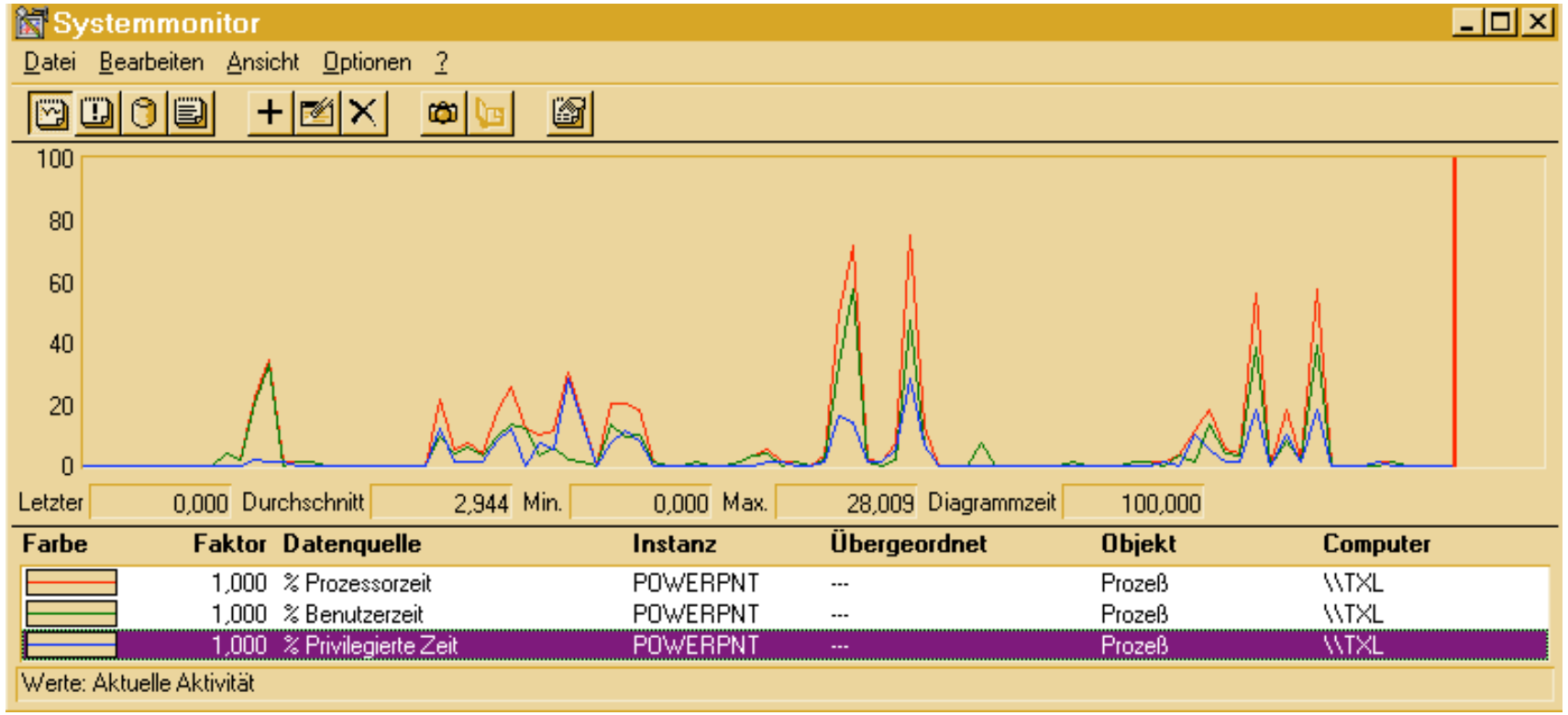
- 2 processor access modes
  - User mode/kernel mode
  - Each page is tagged (access mode)
  - System pages only accessible in kernel mode



Kernel Mode vs. User Mode

- No protection for components running in kernel mode
- Transition from user mode to kernel mode through special instruction (processor changes privilege level)
  - OS traps this instruction and validates arguments to syscalls
  - Transition from user to kernel mode does not affect thread scheduling
- Performance Counters: System/Processor/Process/ Thread – Privileged Time/User time
- Performance Monitor – perfmon.exe

Performance Monitor



Objects and Handles

- Process, thread, file, event objects in Win32 - are mapped on NT executive objects
- Object services read/write object attributes
- Objects:
  - Human-readable names for system resources
  - Resource sharing among processes
  - Resource protection against unauthorized access
- Security/Protection based on NT executive objects
- 2 forms of access control:
  - Discretionary control: read/write/access rights
  - Privileged access: administrator may take ownership of files

Security

- Windows 2000 supports C2-level security (DoD 5200.23-STD, December 1985)
  - Discretionary protection (need-to-know) for shareable system objects (files, directories, processes, threads)
  - Security auditing (accountability of subjects and their actions)
  - Password authentication at logon
  - Prevention of access to un-initialized resources (memory, disk space)
- Windows NT 3.51 was formally evaluated for C2
- Windows NT 4.0 SP 6a passed C2 in December 1999
  - Networked workstation configuration
- European IT Security Criteria FC2/E3 security level

Registry

- System wide software settings: boot & configuration info
- Security database
- Per-user profile settings
- In-memory volatile data (current hardware state)
  - What devices are loaded?
  - Resources used by devices
  - Performance counters are accessed through registry functions
    - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control
    - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
    - HKEY_LOCAL_MACHINE\Software
- Regedt32.exe is the tool to view/modify registry settings

Unicode

- Most internal text strings are stored/processed as 16-bit wide Unicode strings
- Win32 string functions have 2 versions
  - Unicode (wide) version
    - L“This string uses 16-bit characters“
  - ANSI(narrow) version
    - “This string uses 8-bit characters“
  - Generic character representation in Win32
    - _T (“This string uses generic characters“)

(Windows 95/98/ME have Win32 but no Unicode characters,
Windows CE has Win32 but only Unicode characters)

Tools for Viewing Windows 2000 Internals

Tool	Executable	Origin
Performance Monitor	PerfMon	Windows 2000
Registry Editor	RegEdt32	Windows 2000
Windows 2000 Diagnostics	WinMSD	Windows 2000
Kernel Debugger	i386kd, KD, WINDBG	Platform SDK, Windows 2000 DDK
Pool Monitor	poolmon	Windows 2000 CD \Support\Tools
Global Flags	gflags	Windows 2000 CD \Support\Tools
Open Handles	oh	Windows 2000 Resource Kits
QuickSlice	qslice	Windows 2000 Resource Kits
Process Viewer	pviewer, pview	Windows 2000 CD \Support\Tools Platform SDK
Process Explode	pview	www.reskit.com
Process Statistics	pstat	Platform SDK, www.reskit.com
Pool Monitor	poolmon	Windows 2000 CD \Support\Tools, DDK
Object Viewer	WinObj	Platform SD, www.sysinternals.com
Page Fault Monitor	PFMon	Windows 2000 Resource Kits, Platform SDK
Service Control Tool	sc	Windows 2000 Resource Kits
Task (Process) List	tlist	Windows 2000 CD \Support\Tools

www.sysinternals.com

- Windows NT internals articles and tools
  - Many generated using reverse engineering; e.g., no source access
- Some examples:
  - Handlex - show open handles and DLLs by process
  - Listdlls - show DLLs loaded in each process
  - Diskmon/Filemon - log all file I/O operations
  - Regmon - log all registry accesses
  - Winobj - view object manager namespace and objects
- Caveat: Most include a device driver, hence you're added "trusted code"

Sources of Information

- Windows NT Resource Kits
- Platform SDK and Windows NT DDK
  - MSDN Development Platform
- Knowledge Base at www.microsoft.com
- TechNet CD-ROM edition
- Free Builds and Checked Builds
  - Kernel Debuggers
  - I386KD.EXE (command line)
  - WINDBG.EXE (GUI) with platform SDK