Windows Internals

Module 3: System Architecture (Part 1)

Pavel Yosifovich

CTO, CodeValue

pavely@codevalue.net

http://blogs.Microsoft.co.il/blogs/pavely

pluralsight
hardcore developer training

Contents

- Windows design goals
- Windows editions
- General architecture overview
- Function call flow
- Summary

Windows Design Goals

- **Separate address space per process**
  - One process cannot (easily) corrupt another’s memory
- **Protected kernel**
  - User mode applications cannot crash kernel
- **Preemptive multitasking and multithreading**
- **Multiprocessing support**
- **Internationalization support using Unicode**
- **Security throughout the system**
- **Integrated networking**

Windows Design Goals (2)

- **Powerful file system (NTFS)**
  - Supports protection, compression and encryption
- **Run most 16 bit Windows and DOS apps**
  - On 32 bit systems
- **Run POSIX 1003.1 and OS/2 applications**
- **Portable across processors and platforms**
- **Be a great client as well as server platform**

Demo

Unicode in the Windows API

Windows Editions

- **Windows XP Home**
  - □ Designed as a replacement for the Windows 9x/ME family ("Consumer Windows")
- **Windows Professional (2000, XP, Vista, 7, 8)**
  - □ Main desktop (client) OS
- **Windows Server Standard, Advanced, Datacenter editions (Windows 2000, 2003/R2, 2008/R2, 2012)**
  - □ Server platforms
- **Other variants**
  - □ XP starter, XP Home, Media center, Server Web Edition, Home, Premium, Ultimate, Business, Enterprise

Professional vs. Server

- **Same core system files**
- **Differences**
  - Number of processors supported
  - Maximum amount of RAM than can be used
  - Maximum of concurrent network connections supported for file and print sharing
  - Some services only appear in Server versions
  - Other system policies and default settings (e.g. thread quantum)
- **OS type can be discovered by calling GetVersionEx (Win32) or RtlGetVersion(WDK)**

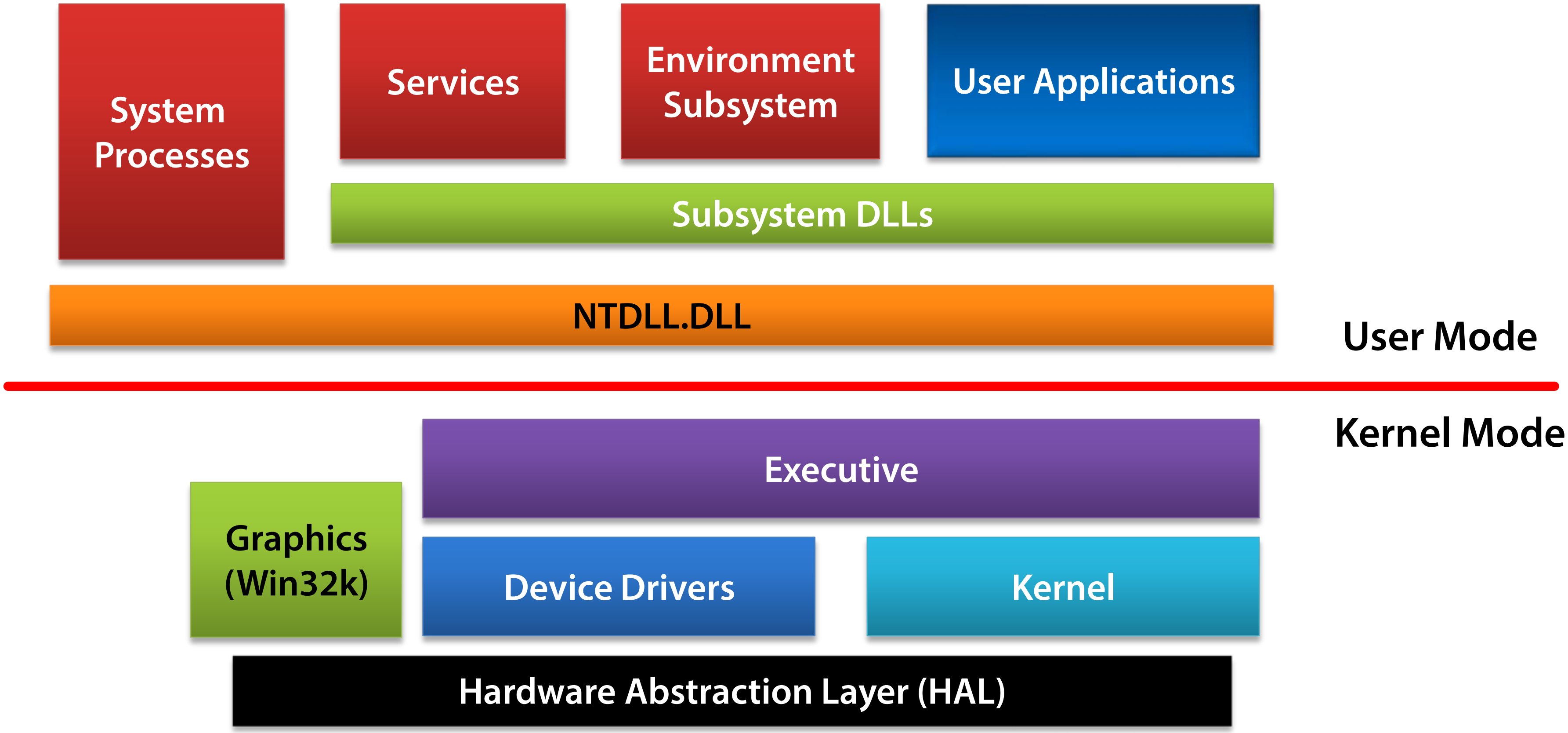
Windows Numeric Versions

- Windows NT 4 (4.0)
- Windows 2000 (5.0)
- Windows XP (5.1)
- Windows Server 2003, 2003 R2 (5.2)
- Windows Vista, Server 2008 (6.0)
- Windows 7, Server 2008 R2 (6.1)
- Windows 8, Server 2012 (6.2)
- Windows 8.1, Server 2012 R2 (6.3)
- These values can be obtained using GetVersionEx (Win32) or RtlGetVersion (WDK)

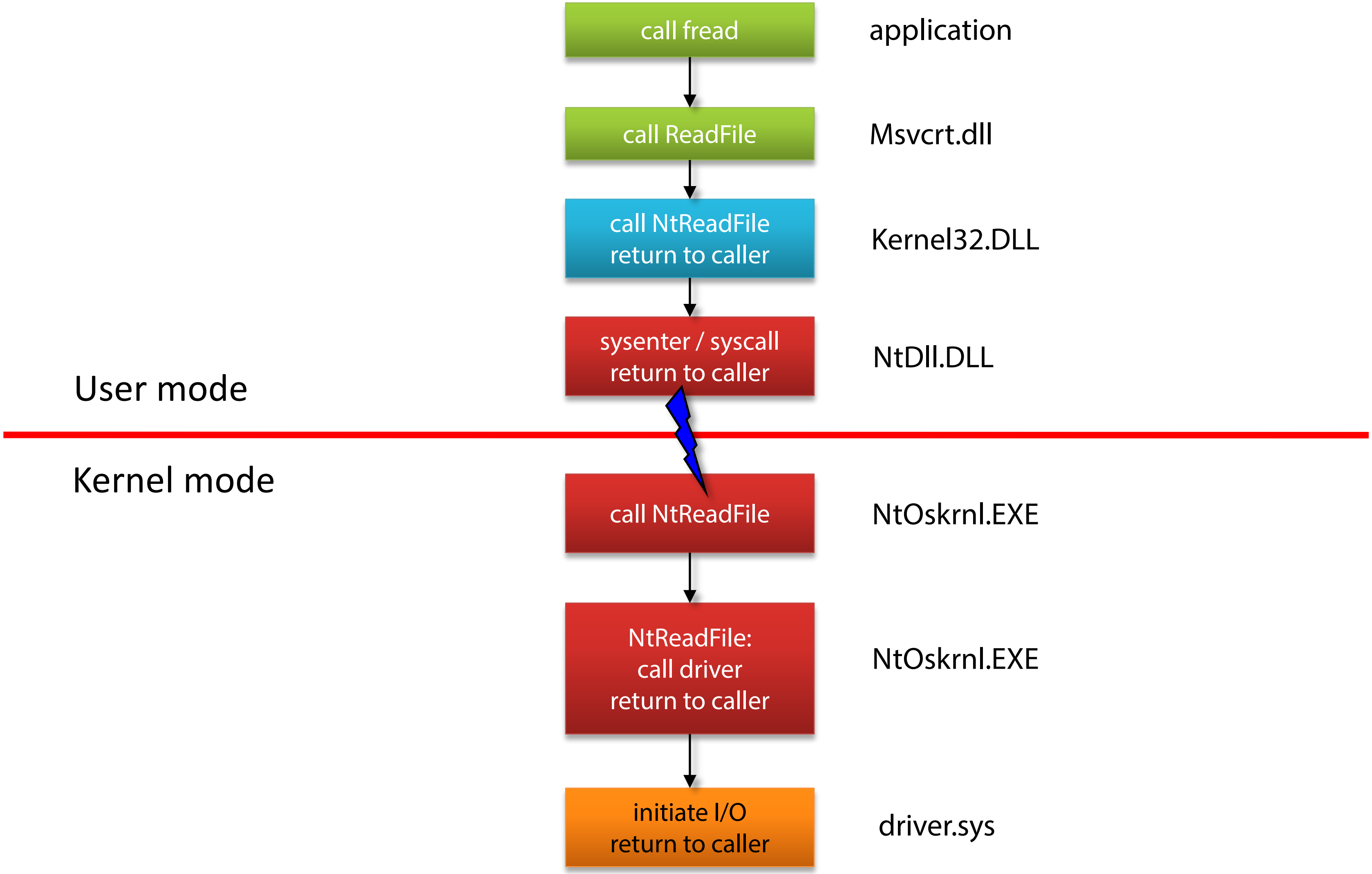
Demo

Looking at a Windows version

General Architecture Overview



Function Call Flow



Brief Overview of WinDbg

- **WinDbg is part of the Debugging Tools for Windows**
- **Other debuggers in the tools: NTSD, CDB, KD**
- **All debuggers are based on the same engine: DbgEng.Dll**
- **NTSD & CDB are user mode debuggers**
  - Practically identical – NTSD spawns a new console window if launched from a console window
- **KD is a kernel mode debugger**
- **WinDbg can serve as a user mode or kernel mode debugger**
- **WinDbg is the only one with a graphical user interface**
- **Most important window is the Command window**
  - Can do anything
  - Some shortcuts available through the menu

Demo

Function call flow

Summary

- Although there are many Windows editions, the kernel is basically the same
- User mode processes use subsystem DLLs to access OS functionality
- A system service call entails transitioning from user mode to kernel mode (and back)