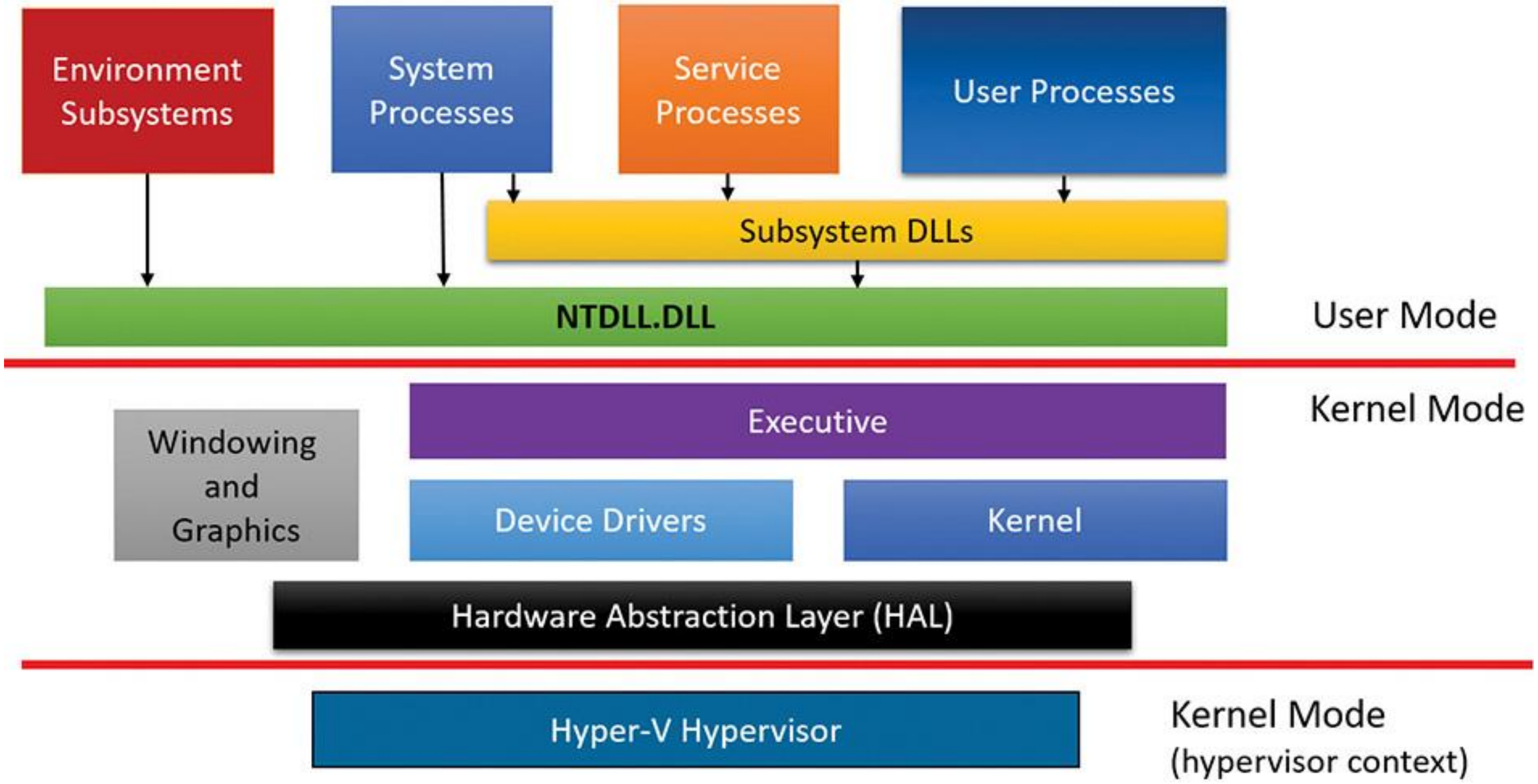
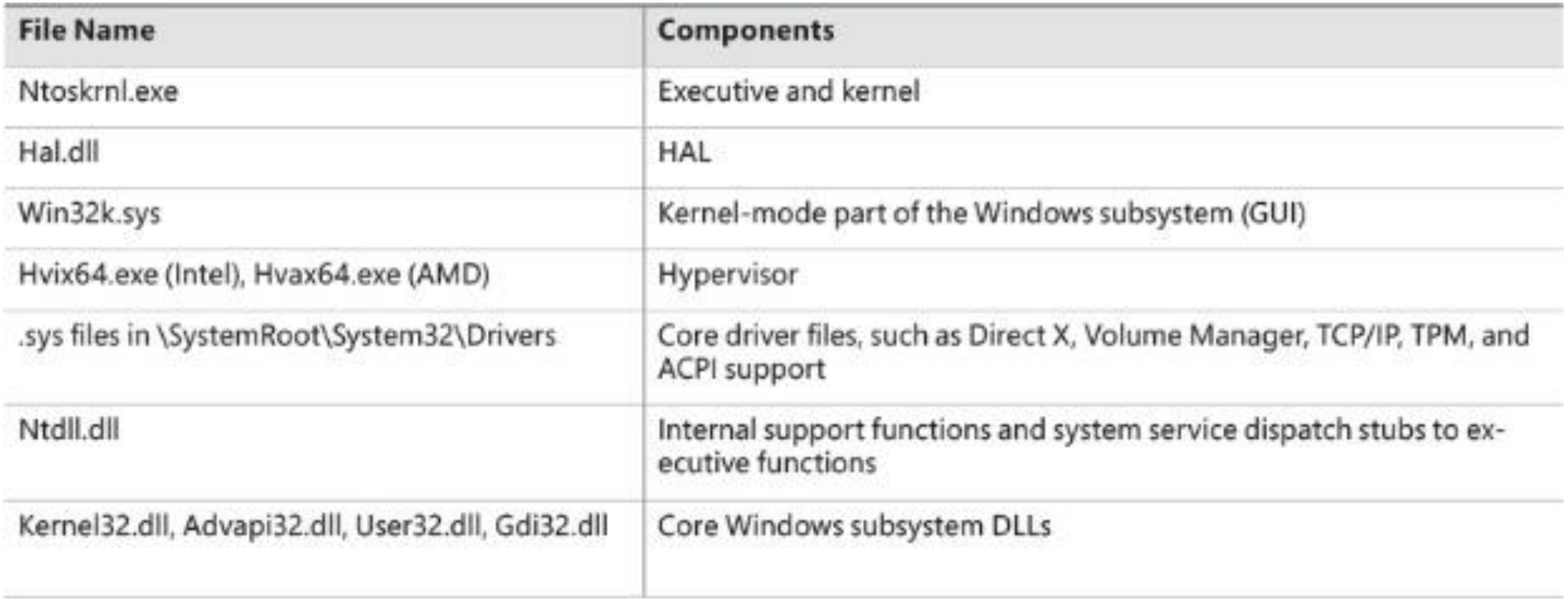
Simplified Windows architecture



Simplified Windows architecture



File Name	Components
Ntoskrnl.exe	Executive and kernel
Hal.dll	HAL
Win32k.sys	Kernel-mode part of the Windows subsystem (GUI)
Hvix64.exe (Intel), Hvax64.exe (AMD)	Hypervisor
.sys files in \SystemRoot\System32\Drivers	Core driver files, such as Direct X, Volume Manager, TCP/IP, TPM, and ACPI support
Ntdll.dll	Internal support functions and system service dispatch stubs to executive functions
Kernel32.dll, Advapi32.dll, User32.dll, Gdi32.dll	Core Windows subsystem DLLs

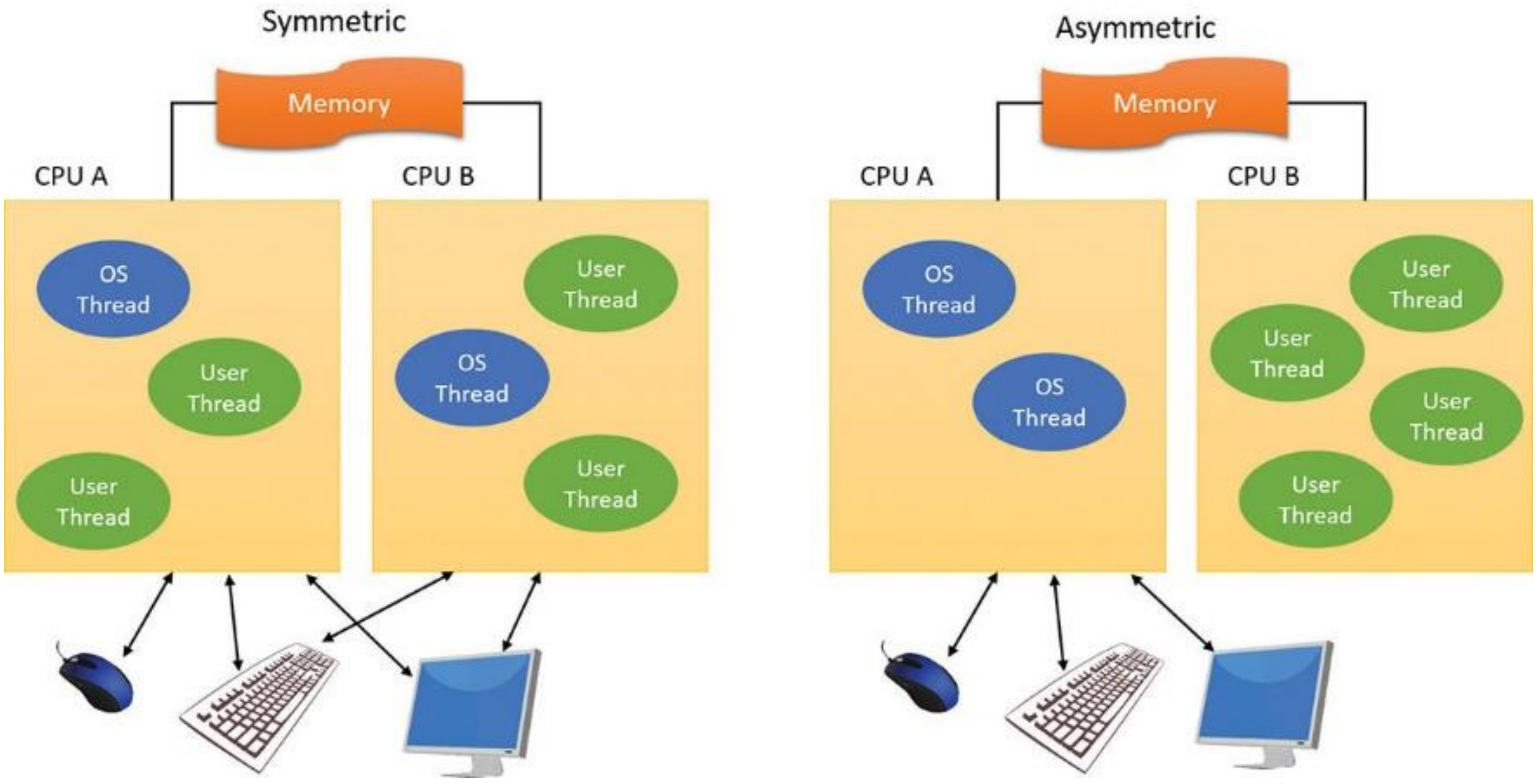
Core Windows System Files

Portability

- Windows was designed to run on a variety of hardware architectures.

- Windows achieves portability across hardware architectures and platforms in two primary ways:
  - By using a layered design
  - By using C

Symmetric multiprocessing



Symmetric multiprocessing

- Windows supports four modern types of multiprocessor systems:
    - Multicore
    - Simultaneous MultiThreaded (SMT)
    - Heterogeneous
    - Non-Uniform Memory Access (NUMA)

Scalability

- Windows incorporates several features that are crucial to its success as a multiprocessor OS:
    - The ability to run OS code on any available processor and on multiple processors at the same time
    - Multiple threads of execution within a single process, each of which can execute simultaneously on different processors
    - Fine-grained synchronization within the kernel (such as spinlocks, queued spinlocks, and pushlocks) as well as within device drivers and server processes, which allows more components to run concurrently on multiple processors
    - Programming mechanisms such as I/O completion ports that facilitate the efficient implementation of multithreaded server processes that can scale well on multiprocessor systems

Differences between client and server versions

- There are six desktop client versions of Windows 10:
  - Windows 10 Home
  - Windows 10 Pro
  - Windows 10 Education
  - Windows 10 Pro Education
  - Windows 10 Enterprise
  - Windows 10 Enterprise Long Term Servicing Branch (LTSB)

Differences between client and server versions

- There are six different versions of Windows Server 2016:
  - Windows Server 2016 Datacenter
  - Windows Server 2016 Standard
  - Windows Server 2016 Essentials
  - Windows Server 2006 MultiPoint Premium Server
  - Windows Storage Server 2016
  - Microsoft Hyper-V Server 2016

Differences between client and server versions

- Other non-desktop editions include:
  - Windows 10 Mobile
  - Windows 10 Mobile Enterprise
  - Windows 10 IoT Core
  - IoT Core Enterprise
  - IoT Mobile Enterprise

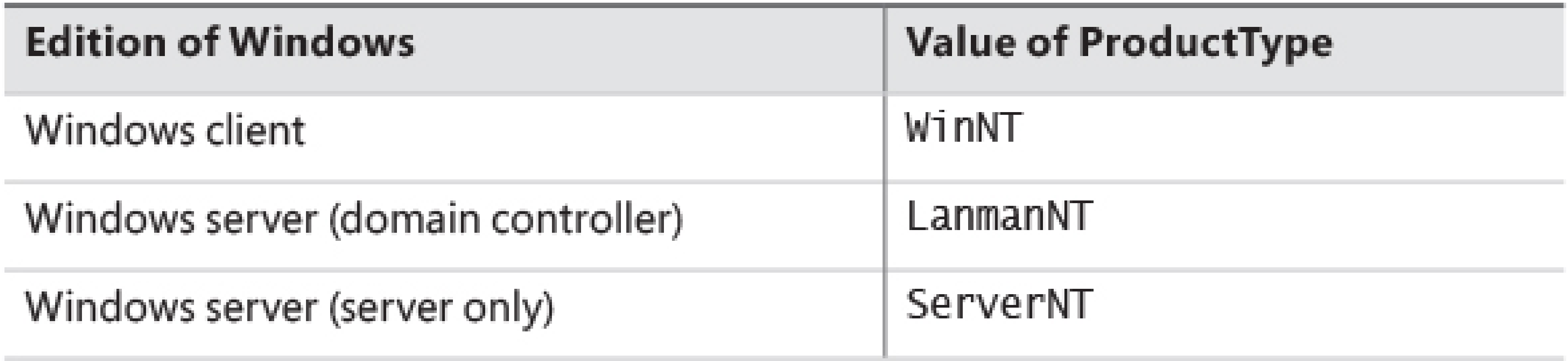
Differences between client and server versions

These versions differ as follows:

- Core-based (rather than socket-based) pricing for the Server 2016 Datacenter and Standard edition
- The number of total logical processors supported
- For server systems, the number of Hyper-V containers allowed to run (client systems support only namespace-based Windows containers)
- The amount of physical memory supported (actually highest physical address usable for RAM)
- The number of concurrent network connections supported (for example, a maximum of 10 concurrent connections are allowed to the file and print services in client versions)
- Support for multi-touch and Desktop Composition
- Support for features such as BitLocker, VHD booting, AppLocker, Hyper-V, and more than 100 other configurable licensing policy values
- Layered services that come with Windows Server editions that don’t come with the client editions(for example, directory services, Host Guardian, Storage Spaces Direct, shielded virtual machines, and clustering)

Differences between client and server versions

- How does the system know which edition is?
  - By querying the registry values `ProductType`, `ProductSuite` and `ProductPolicy` under the ”HKLM\SYSTEM\CurrentControlSet\Control\ProductOptions” key



Edition of Windows	Value of ProductType
Windows client	WinNT
Windows server (domain controller)	LanmanNT
Windows server (server only)	ServerNT

- From user-mode: `VerifyVersionInfo`
- From device deriver: `RtlGetVersion` and `RtlVerifyVersionInfo`

Differences between client and server versions

How do the systems differ in operation?

- Server systems are optimized by default for system throughput as high performance application servers, whereas the client version (although it has server capabilities) is optimized for response time for interactive desktop use.
    - System boot time decisions: such as the size and number of OS heaps (or pools), the number of internal system worker threads, and the size of the system data cache
    - Run-time policy decisions: such as the way the memory manager trades off system and process memory demands and the default length of the time slice, or thread quantum

Checked build

- A special internal debug version of Windows (available only for Windows 8.1 and earlier).
- A recompilation of the Windows source code with a compile-time flag defined called DBG, which causes compile time, conditional debugging, and tracing code to be included.
- The post-processing of the Windows binaries to optimize code layout for faster execution is not performed, to make it easier to understand the machine code.
- To aid device driver developers because it performs more stringent error-checking on kernel-mode functions called by device drivers or other system code.
- Because a full checked build was often unstable and impossible to run in most environments, Microsoft provides a checked kernel and HAL only for Windows 10 and later.