A **hypervisor** (or **virtual machine monitor**, **VMM**, **virtualizer**) is computer [software](https://en.wikipedia.org/wiki/Software), [firmware](https://en.wikipedia.org/wiki/Firmware) or [hardware](https://en.wikipedia.org/wiki/Computer_hardware) that creates and runs [virtual machines](https://en.wikipedia.org/wiki/Virtual_machine). A computer on which a hypervisor runs one or more virtual machines is called a *host machine*, and each virtual machine is called a *guest machine*. The hypervisor presents the guest operating systems with a [virtual operating platform](https://en.wikipedia.org/wiki/Platform_virtualization) and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, [Linux](https://en.wikipedia.org/wiki/Linux), [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), and [macOS](https://en.wikipedia.org/wiki/MacOS) instances can all run on a single physical [x86](https://en.wikipedia.org/wiki/X86) machine. This contrasts with [operating-system-level virtualization](https://en.wikipedia.org/wiki/Operating-system-level_virtualization), where all instances (usually called *containers*) must share a single kernel, though the guest operating systems can differ in [user space](https://en.wikipedia.org/wiki/User_space), such as different [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution) with the same kernel.

The term *hypervisor* is a variant of *supervisor*, a traditional term for the [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) of an [operating system](https://en.wikipedia.org/wiki/Operating_system): the hypervisor is the supervisor of the supervisors,[[1]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-1) with [*hyper-*](https://en.wiktionary.org/wiki/hyper-) used as a stronger variant of [*super-*](https://en.wiktionary.org/wiki/super-).[[a]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-2) The term dates to circa 1970;[[2]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-3) in the earlier [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) (1967) system, the term *Control Program* was used instead.

Classification

In their 1974 article, *Formal Requirements for Virtualizable Third Generation Architectures*, [Gerald J. Popek](https://en.wikipedia.org/wiki/Gerald_J._Popek) and [Robert P. Goldberg](https://en.wikipedia.org/wiki/Robert_P._Goldberg) classified two types of hypervisor:[[3]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-4)

**Type-1, native or bare-metal hypervisors**

These hypervisors run directly on the host's hardware to control the hardware and to manage guest operating systems. For this reason, they are sometimes called [bare metal](https://en.wikipedia.org/wiki/Bare_machine) hypervisors. The first hypervisors, which IBM developed in the 1960s, were native hypervisors.[[4]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-5) These included the test software [SIMMON](https://en.wikipedia.org/wiki/SIMMON) and the [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) operating system (the predecessor of IBM's [z/VM](https://en.wikipedia.org/wiki/Z/VM)). Modern equivalents include AntsleOS,[[5]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-6) Microsoft [Hyper-V](https://en.wikipedia.org/wiki/Hyper-V) and [Xbox One system software](https://en.wikipedia.org/wiki/Xbox_One_system_software), Nutanix [AHV](https://www.nutanix.com/products/acropolis/virtualization), [XCP-ng](https://xcp-ng.org/), [Oracle VM Server for SPARC](https://en.wikipedia.org/wiki/Oracle_VM_Server_for_SPARC), [Oracle VM Server for x86](https://en.wikipedia.org/wiki/Oracle_VM_Server_for_x86), [POWER Hypervisor](https://en.wikipedia.org/w/index.php?title=POWER_Hypervisor&action=edit&redlink=1)[[6]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-7) and [VMware ESXi](https://en.wikipedia.org/wiki/VMware_ESXi) (formerly ESX) and [Xen](https://en.wikipedia.org/wiki/Xen).

**Type-2 or hosted hypervisors**

These hypervisors run on a conventional operating system (OS) just as other computer programs do. A guest operating system runs as a [process](https://en.wikipedia.org/wiki/Computer_process) on the host. Type-2 hypervisors abstract guest operating systems from the host operating system. [Parallels Desktop for Mac](https://en.wikipedia.org/wiki/Parallels_Desktop_for_Mac), [QEMU](https://en.wikipedia.org/wiki/QEMU), [VirtualBox](https://en.wikipedia.org/wiki/VirtualBox), [VMware Player](https://en.wikipedia.org/wiki/VMware_Player) and [VMware Workstation](https://en.wikipedia.org/wiki/VMware_Workstation) are examples of type-2 hypervisors.

The distinction between these two types is not always clear. For instance, Linux's [Kernel-based Virtual Machine](https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine) (KVM) and [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD)'s [bhyve](https://en.wikipedia.org/wiki/Bhyve) are [kernel modules](https://en.wikipedia.org/wiki/Kernel_module)[[7]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-8) that effectively convert the host operating system to a type-1 hypervisor.[[8]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-IowaState-9) At the same time, since [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution) and FreeBSD are still general-purpose operating systems, with applications competing with each other for VM resources, KVM and bhyve can also be categorized as type-2 hypervisors.[[9]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-10)

## Mainframe origins[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=2)]

The first hypervisors providing [full virtualization](https://en.wikipedia.org/wiki/Full_virtualization) were the test tool [SIMMON](https://en.wikipedia.org/wiki/SIMMON) and IBM's one-off research [CP-40](https://en.wikipedia.org/wiki/IBM_CP-40) system, which began production use in January 1967, and became the first version of IBM's [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) operating system. CP-40 ran on a [S/360-40](https://en.wikipedia.org/wiki/System/360) that was modified at the IBM [Cambridge Scientific Center](https://en.wikipedia.org/wiki/Cambridge_Scientific_Center) to support dynamic address translation, a feature that enabled virtualization. Prior to this time, computer hardware had only been virtualized to the extent to allow multiple user applications to run concurrently, such as in [CTSS](https://en.wikipedia.org/wiki/Compatible_Time-Sharing_System) and [IBM M44/44X](https://en.wikipedia.org/wiki/IBM_M44/44X). With CP-40, the hardware's *supervisor state* was virtualized as well, allowing multiple operating systems to run concurrently in separate [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine) contexts.

Programmers soon implemented CP-40 (as [CP-67](https://en.wikipedia.org/wiki/CP-67)) for the [IBM System/360-67](https://en.wikipedia.org/wiki/IBM_System/360-67), the first production computer system capable of [full virtualization](https://en.wikipedia.org/wiki/Full_virtualization). IBM first shipped this machine in 1966; it included page-translation-table hardware for virtual memory, and other techniques that allowed a [full virtualization](https://en.wikipedia.org/wiki/Full_virtualization) of all kernel tasks, including I/O and interrupt handling. (Note that its "official" operating system, the ill-fated [TSS/360](https://en.wikipedia.org/wiki/TSS/360), did not employ full virtualization.) Both CP-40 and CP-67 began production use in 1967. [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) was available to IBM customers from 1968 to early 1970s, in source code form without support.

[CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) formed part of IBM's attempt to build robust [time-sharing](https://en.wikipedia.org/wiki/Time-sharing) systems for its [mainframe](https://en.wikipedia.org/wiki/Mainframe_computer) computers. By running multiple operating systems concurrently, the hypervisor increased system robustness and stability: Even if one operating system crashed, the others would continue working without interruption. Indeed, this even allowed [beta](https://en.wikipedia.org/wiki/Development_stage#Beta) or experimental versions of operating systems‍—‌or even of new hardware[[10]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-11)‍—‌to be deployed and debugged, without jeopardizing the stable main production system, and without requiring costly additional development systems.

IBM announced its [System/370](https://en.wikipedia.org/wiki/System/370) series in 1970 without the [virtual memory](https://en.wikipedia.org/wiki/Virtual_memory) feature needed for virtualization, but added it in the August 1972 Advanced Function announcement. Virtualization has been featured in all successor systems (all modern-day IBM mainframes, such as the [zSeries](https://en.wikipedia.org/wiki/ZSeries) line, retain backward compatibility with the 1960s-era IBM S/360 line). The 1972 announcement also included [VM/370](https://en.wikipedia.org/wiki/VM_(operating_system)), a reimplementation of [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) for the S/370. Unlike [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS), IBM provided support for this version (though it was still distributed in source code form for several releases). VM stands for [*Virtual Machine*](https://en.wikipedia.org/wiki/Virtual_Machine), emphasizing that all, and not just some, of the hardware interfaces are virtualized. Both VM and CP/CMS enjoyed early acceptance and rapid development by universities, corporate users, and [time-sharing](https://en.wikipedia.org/wiki/Time-sharing) vendors, as well as within IBM. Users played an active role in ongoing development, anticipating trends seen in modern [open source](https://en.wikipedia.org/wiki/Open-source_model) projects. However, in a series of disputed and bitter battles, time-sharing lost out to [batch processing](https://en.wikipedia.org/wiki/Batch_processing) through IBM political infighting, and VM remained IBM's "other" mainframe operating system for decades, losing to [MVS](https://en.wikipedia.org/wiki/MVS). It enjoyed a resurgence of popularity and support from 2000 as the [z/VM](https://en.wikipedia.org/wiki/Z/VM) product, for example as the platform for [Linux on IBM Z](https://en.wikipedia.org/wiki/Linux_on_IBM_Z).

As mentioned above, the VM control program includes a *hypervisor-call* handler that intercepts DIAG ("Diagnose", opcode x'83') instructions used within a virtual machine. This provides fast-path non-virtualized execution of file-system access and other operations (DIAG is a model-dependent privileged instruction, not used in normal programming, and thus is not virtualized. It is therefore available for use as a signal to the "host" operating system). When first implemented in [CP/CMS](https://en.wikipedia.org/wiki/CP/CMS) release 3.1, this use of DIAG provided an operating system interface that was analogous to the [System/360](https://en.wikipedia.org/wiki/System/360) [Supervisor Call instruction](https://en.wikipedia.org/wiki/Supervisor_Call_instruction) (SVC), but that did not require altering or extending the system's virtualization of SVC.

In 1985 IBM introduced the [PR/SM](https://en.wikipedia.org/wiki/PR/SM) hypervisor to manage [logical partitions](https://en.wikipedia.org/wiki/Logical_partition_(virtual_computing_platform)) (LPAR).

## Operating system support[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=3)]

Several factors led to a resurgence around 2005 in the use of [virtualization](https://en.wikipedia.org/wiki/Platform_virtualization) technology among [Unix](https://en.wikipedia.org/wiki/Unix), [Linux](https://en.wikipedia.org/wiki/Linux), and other [Unix-like](https://en.wikipedia.org/wiki/Unix-like) operating systems:[[11]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-12)

* Expanding hardware capabilities, allowing each single machine to do more simultaneous work
* Efforts to control costs and to simplify management through consolidation of servers
* The need to control large [multiprocessor](https://en.wikipedia.org/wiki/Multiprocessor) and [cluster](https://en.wikipedia.org/wiki/Computer_cluster) installations, for example in [server farms](https://en.wikipedia.org/wiki/Server_farm) and [render farms](https://en.wikipedia.org/wiki/Render_farm)
* The improved security, reliability, and device independence possible from hypervisor architectures
* The ability to run complex, OS-dependent applications in different hardware or OS environments

Major Unix vendors, including [HP](https://en.wikipedia.org/wiki/Hewlett-Packard), [IBM](https://en.wikipedia.org/wiki/IBM), [SGI](https://en.wikipedia.org/wiki/Silicon_Graphics) and [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems), have been selling virtualized hardware since before 2000. These have generally been large, expensive systems (in the multimillion-dollar range at the high end), although virtualization has also been available on some low- and mid-range systems, such as IBM's [pSeries](https://en.wikipedia.org/wiki/System_p) servers, [HP Superdome](https://en.wikipedia.org/wiki/HP_Superdome) series machines and [Sun](https://en.wikipedia.org/wiki/Sun_Microsystems)/[Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)'s T-series CoolThreads servers.

Although [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) has always been the only guest domain OS officially supported by Sun/Oracle on their [Logical Domains](https://en.wikipedia.org/wiki/Logical_Domains) hypervisor, as of late 2006, [Linux](https://en.wikipedia.org/wiki/Linux) (Ubuntu and Gentoo), and [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) have been ported to run on top of the hypervisor (and can all run simultaneously on the same processor, as fully virtualized independent guest OSes). Wind River "[Carrier Grade Linux](https://en.wikipedia.org/wiki/Carrier_Grade_Linux)" also runs on Sun's Hypervisor.[[12]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-13) Full virtualization on [SPARC](https://en.wikipedia.org/wiki/SPARC) processors proved straightforward: since its inception in the mid-1980s Sun deliberately kept the SPARC architecture clean of artifacts that would have impeded virtualization. (Compare with virtualization on x86 processors below.)[[13]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-14)

HPE provides [HP Integrity Virtual Machines](https://en.wikipedia.org/wiki/HP_Integrity_Virtual_Machines) (Integrity VM) to host multiple operating systems on their [Itanium](https://en.wikipedia.org/wiki/Itanium) powered Integrity systems. Itanium can run [HP-UX](https://en.wikipedia.org/wiki/HP-UX), Linux, Windows and [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS), and these environments are also supported as virtual servers on HP's Integrity VM platform. The HP-UX operating system hosts the Integrity VM hypervisor layer that allows for many important features of HP-UX to be taken advantage of and provides major differentiation between this platform and other commodity platforms - such as processor hotswap, memory hotswap, and dynamic kernel updates without system reboot. While it heavily leverages HP-UX, the Integrity VM hypervisor is really a hybrid that runs on bare-metal while guests are executing. Running normal HP-UX applications on an Integrity VM host is heavily discouraged,[[*by whom?*](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Words_to_watch#Unsupported_attributions)] because Integrity VM implements its own memory management, scheduling and I/O policies that are tuned for virtual machines and are not as effective for normal applications. HPE also provides more rigid partitioning of their Integrity and HP9000 systems by way of VPAR and [nPar](https://en.wikipedia.org/wiki/HP_nPar_(Hard_Partitioning)) technology, the former offering shared resource partitioning and the latter offering complete I/O and processing isolation. The flexibility of virtual server environment (VSE) has given way to its use more frequently in newer deployments.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

IBM provides virtualization partition technology known as [logical partitioning](https://en.wikipedia.org/wiki/LPAR) (LPAR) on [System/390](https://en.wikipedia.org/wiki/System/390), [zSeries](https://en.wikipedia.org/wiki/ZSeries), [pSeries](https://en.wikipedia.org/wiki/PSeries) and [iSeries](https://en.wikipedia.org/wiki/ISeries) systems. For IBM's Power Systems, the POWER Hypervisor (PHYP) is a native (bare-metal) hypervisor in firmware and provides isolation between LPARs. Processor capacity is provided to LPARs in either a dedicated fashion or on an entitlement basis where unused capacity is harvested and can be re-allocated to busy workloads. Groups of LPARs can have their processor capacity managed as if they were in a "pool" - IBM refers to this capability as Multiple Shared-Processor Pools (MSPPs) and implements it in servers with the [POWER6](https://en.wikipedia.org/wiki/POWER6) processor. LPAR and MSPP capacity allocations can be dynamically changed. Memory is allocated to each LPAR (at LPAR initiation or dynamically) and is address-controlled by the POWER Hypervisor. For real-mode addressing by operating systems (AIX, Linux, IBM i), the [POWER](https://en.wikipedia.org/wiki/IBM_POWER_microprocessors) processors ([POWER4](https://en.wikipedia.org/wiki/POWER4) onwards) have designed virtualization capabilities where a hardware address-offset is evaluated with the OS address-offset to arrive at the physical memory address. Input/Output (I/O) adapters can be exclusively "owned" by LPARs or shared by LPARs through an appliance partition known as the Virtual I/O Server (VIOS). The Power Hypervisor provides for high levels of reliability, availability and serviceability (RAS) by facilitating hot add/replace of many parts (model dependent: processors, memory, I/O adapters, blowers, power units, disks, system controllers, etc.)

Similar trends have occurred with x86/x86-64 server platforms, where [open-source](https://en.wikipedia.org/wiki/Open-source_model) projects such as [Xen](https://en.wikipedia.org/wiki/Xen) have led virtualization efforts. These include hypervisors built on Linux and Solaris kernels as well as custom kernels. Since these technologies span from large systems down to desktops, they are described in the next section.

x86 systems[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=4)]

*Main article:*[*x86 virtualization*](https://en.wikipedia.org/wiki/X86_virtualization)

Starting in 2005, CPU vendors have added hardware virtualization assistance to their products, for example: [Intel VT-x](https://en.wikipedia.org/wiki/Intel_VT-x) (code-named Vanderpool) and [AMD-V](https://en.wikipedia.org/wiki/AMD-V) (code-named Pacifica).

An alternative approach requires modifying the guest operating-system to make system calls to the hypervisor, rather than executing machine I/O instructions that the hypervisor simulates. This is called [paravirtualization](https://en.wikipedia.org/wiki/Paravirtualization) in [Xen](https://en.wikipedia.org/wiki/Xen), a "hypercall" in [Parallels Workstation](https://en.wikipedia.org/wiki/Parallels_Workstation), and a "DIAGNOSE code" in IBM's [VM](https://en.wikipedia.org/wiki/VM_(operating_system)). All are really the same thing, a system call to the underlying hypervisor. Some microkernels such as [Mach](https://en.wikipedia.org/wiki/Mach_microkernel) and [L4](https://en.wikipedia.org/wiki/L4_microkernel_family) are flexible enough such that "[paravirtualization](https://en.wikipedia.org/wiki/Paravirtualization)" of guest operating systems is possible.

Embedded systems[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=5)]

[Embedded hypervisors](https://en.wikipedia.org/wiki/Embedded_hypervisor), targeting [embedded systems](https://en.wikipedia.org/wiki/Embedded_system) and certain [real-time operating system](https://en.wikipedia.org/wiki/Real-time_operating_system) (RTOS) environments, are designed with different requirements when compared to desktop and enterprise systems, including robustness, security and [real-time](https://en.wikipedia.org/wiki/Real-time_computing) capabilities. The resource-constrained nature of many embedded systems, especially battery-powered mobile systems, imposes a further requirement for small memory-size and low overhead. Finally, in contrast to the ubiquity of the x86 architecture in the PC world, the embedded world uses a wider variety of architectures and less standardized environments. Support for virtualization requires [memory protection](https://en.wikipedia.org/wiki/Memory_protection) (in the form of a [memory management unit](https://en.wikipedia.org/wiki/Memory_management_unit) or at least a memory protection unit) and a distinction between [user mode](https://en.wikipedia.org/wiki/User_mode) and [privileged mode](https://en.wikipedia.org/wiki/Privileged_mode), which rules out most [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller). This still leaves [x86](https://en.wikipedia.org/wiki/X86), [MIPS](https://en.wikipedia.org/wiki/MIPS_architecture), [ARM](https://en.wikipedia.org/wiki/ARM_architecture) and [PowerPC](https://en.wikipedia.org/wiki/PowerPC) as widely deployed architectures on medium- to high-end embedded systems.[[14]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-15)

As manufacturers of embedded systems usually have the source code to their operating systems, they have less need for [full virtualization](https://en.wikipedia.org/wiki/Full_virtualization) in this space. Instead, the performance advantages of [paravirtualization](https://en.wikipedia.org/wiki/Paravirtualization) make this usually the virtualization technology of choice. Nevertheless, ARM and MIPS have recently added full virtualization support as an IP option and has included it in their latest high-end processors and architecture versions, such as [ARM Cortex-A15 MPCore](https://en.wikipedia.org/wiki/ARM_Cortex-A15_MPCore) and ARMv8 EL2.

Other differences between virtualization in server/desktop and embedded environments include requirements for efficient sharing of resources across virtual machines, high-bandwidth, low-latency inter-VM communication, a global view of scheduling and power management, and fine-grained control of information flows.[[15]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-Hei08-16)

Security implications[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=6)]

*Main article:*[*Hyperjacking*](https://en.wikipedia.org/wiki/Hyperjacking)

The use of hypervisor technology by [malware](https://en.wikipedia.org/wiki/Malware) and [rootkits](https://en.wikipedia.org/wiki/Rootkit) installing themselves as a hypervisor below the operating system, known as [*hyperjacking*](https://en.wikipedia.org/wiki/Hyperjacking), can make them more difficult to detect because the malware could intercept any operations of the operating system (such as someone entering a password) without the anti-malware software necessarily detecting it (since the malware runs below the entire operating system). Implementation of the concept has allegedly occurred in the [SubVirt](https://en.wikipedia.org/w/index.php?title=SubVirt&action=edit&redlink=1) laboratory rootkit (developed jointly by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) and [University of Michigan](https://en.wikipedia.org/wiki/University_of_Michigan) researchers[[16]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-17)) as well as in the [Blue Pill malware](https://en.wikipedia.org/wiki/Blue_Pill_(software)) package. However, such assertions have been disputed by others who claim that it would be possible to detect the presence of a hypervisor-based rootkit.[[17]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-18)

In 2009, researchers from Microsoft and [North Carolina State University](https://en.wikipedia.org/wiki/North_Carolina_State_University) demonstrated a hypervisor-layer anti-rootkit called [Hooksafe](https://en.wikipedia.org/wiki/Hooksafe) that can provide generic protection against kernel-mode [rootkits](https://en.wikipedia.org/wiki/Rootkit).[[18]](https://en.wikipedia.org/wiki/Hypervisor#cite_note-19)

Notes[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=7)]

* 1. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-2) *super-* is from Latin, meaning "above", while *hyper-* is from the [cognate](https://en.wikipedia.org/wiki/Cognate) term in [Ancient Greek](https://en.wikipedia.org/wiki/Ancient_Greek) (ὑπέρ), also meaning *above* or *over*.

References[[edit](https://en.wikipedia.org/w/index.php?title=Hypervisor&action=edit&section=8)]

* 1. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-1) *Bernard Golden (2011). Virtualization For Dummies. p.*[*54*](https://books.google.com/books?id=2ppZkdmpSlgC&pg=PA54)*.*
  2. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-3) [*"How did the term "hypervisor" come into use?"*](http://programmers.stackexchange.com/questions/196405/how-did-the-term-hypervisor-come-into-use)*.*
  3. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-4) *Popek, Gerald J.; Goldberg, Robert P. (1974). "Formal requirements for virtualizable third generation architectures". Communications of the ACM.****17****(7): 412–421.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1145/361011.361073*](https://doi.org/10.1145%2F361011.361073)*.*
  4. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-5) *Meier, Shannon (2008).*[*"IBM Systems Virtualization: Servers, Storage, and Software"*](http://www.redbooks.ibm.com/redpapers/pdfs/redp4396.pdf)*(PDF). pp. 2, 15, 20. Retrieved December 22, 2015.*
  5. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-6) [*"antsleOS Architecture - antsle Docs"*](http://docs.antsle.com/arch/)*. docs.antsle.com. Retrieved June 28, 2018.*
  6. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-7) <https://www.ibm.com/support/knowledgecenter/POWER6/iphb2/iphb2hypervisor.htm>
  7. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-8) *Dexter, Michael.*[*"Hands-on bhyve"*](http://callfortesting.org/bhyve/)*. CallForTesting.org. Retrieved September 24, 2013.*
  8. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-IowaState_9-0) *Graziano, Charles (2011).*[*"A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project"*](http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=3243&context=etd)*. Graduate Theses and Dissertations. Iowa State University. Retrieved January 29, 2013.*
  9. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-10) *Pariseau, Beth (April 15, 2011).*[*"KVM reignites Type 1 vs. Type 2 hypervisor debate"*](http://searchservervirtualization.techtarget.com/news/2240034817/KVM-reignites-Type-1-vs-Type-2-hypervisor-debate)*. SearchServerVirtualization. TechTarget. Retrieved January 29, 2013.*
  10. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-11) See [History of CP/CMS](https://en.wikipedia.org/wiki/History_of_CP/CMS) for virtual-hardware simulation in the development of the [System/370](https://en.wikipedia.org/wiki/System/370)
  11. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-12) *Loftus, Jack (December 19, 2005).*[*"Xen virtualization quickly becoming open source 'killer app'"*](http://searchdatacenter.techtarget.com/news/1153127/Xen-virtualization-quickly-becoming-open-source-killer-app)*. TechTarget. Retrieved October 26, 2015.*
  12. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-13) [*"Wind River To Support Sun's Breakthrough UltraSPARC T1 Multithreaded Next-Generation Processor"*](http://www.windriver.com/news/press/pr.html?ID=3881)*. Wind River Newsroom (Press release). Alameda, California. November 1, 2006. Retrieved October 26, 2015.*
  13. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-14) *Fritsch, Lothar; Husseiki, Rani; Alkassar, Ammar.*[*Complementary and Alternative Technologies to Trusted Computing (TC-Erg./-A.), Part 1, A study on behalf of the German Federal Office for Information Security (BSI)*](https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/TC_ErgA/TC-ErgA_Part1.pdf?__blob=publicationFile)*(PDF) (Report).*
  14. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-15) *Strobl, Marius (2013).*[*Virtualization for Reliable Embedded Systems*](http://www.grin.com/e-book/233001/)*. Munich: GRIN Publishing GmbH. pp. 5–6.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-3-656-49071-5*](https://en.wikipedia.org/wiki/Special:BookSources/978-3-656-49071-5)*. Retrieved March 7, 2015.*
  15. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-Hei08_16-0) [*Gernot Heiser*](https://en.wikipedia.org/wiki/Gernot_Heiser)*(April 2008).*[*"The role of virtualization in embedded systems"*](https://web.archive.org/web/20120321212358/https:/ertos.nicta.com.au/publications/papers/Heiser_08.abstract)*. Proc. 1st Workshop on Isolation and Integration in Embedded Systems (IIES'08). pp. 11–16. Archived from*[*the original*](http://ertos.nicta.com.au/publications/papers/Heiser_08.abstract)*on March 21, 2012. Retrieved April 8, 2009.*
  16. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-17) [*"SubVirt: Implementing malware with virtual machines"*](http://web.eecs.umich.edu/~pmchen/papers/king06.pdf)*(PDF).*[*University of Michigan*](https://en.wikipedia.org/wiki/University_of_Michigan)*,*[*Microsoft*](https://en.wikipedia.org/wiki/Microsoft)*. April 3, 2006. Retrieved September 15, 2008.*
  17. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-18) [*"Debunking Blue Pill myth"*](http://www.virtualization.info/2006/08/debunking-blue-pill-myth.html)*. Virtualization.info. August 11, 2006. Retrieved December 10,2010.*
  18. [**^**](https://en.wikipedia.org/wiki/Hypervisor#cite_ref-19) *Wang, Zhi; Jiang, Xuxian; Cui, Weidong; Ning, Peng (August 11, 2009).*[*Countering Kernel Rootkits with Lightweight Hook Protection*](http://research.microsoft.com/en-us/um/people/wdcui/papers/hooksafe-ccs09.pdf)*(PDF). Proceedings of the 16th ACM Conference on Computer and Communications Security. CCS '09. Chicago, Illinois, USA:*[*ACM*](https://en.wikipedia.org/wiki/Association_for_Computing_Machinery)*.*[*CiteSeerX*](https://en.wikipedia.org/wiki/CiteSeerX_(identifier))[*10.1.1.147.9928*](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.9928)*.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1145/1653662.1653728*](https://doi.org/10.1145%2F1653662.1653728)*.*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-1-60558-894-0*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-60558-894-0)*. Retrieved November 11, 2009.*