**IT5423**

**Module 1: Introduction to Operating Systems**

**Learning objectives**:

* Describe types and components of operating systems
* Apply commands in Linux OS

Contents

Contents

[**Operating systems** 2](#_Toc500162218)

[**Types of operating systems** 3](#_Toc500162219)

[Single- and multi-tasking 3](#_Toc500162220)

[Single- and multi-user 4](#_Toc500162221)

[Distributed 4](#_Toc500162222)

[Templated 4](#_Toc500162223)

[Embedded 4](#_Toc500162224)

[Real-time 4](#_Toc500162225)

[**History of OS** 4](#_Toc500162226)

[**Mainframes** 5](#_Toc500162227)

[**Microcomputers** 6](#_Toc500162228)

[**Unix and Unix-like operating systems** 7](#_Toc500162229)

[**BSD and its descendants** 8](#_Toc500162230)

[**macOS** 8](#_Toc500162231)

[**Linux** 9](#_Toc500162232)

[**Microsoft Windows** 10](#_Toc500162233)

[**Components of OS** 10](#_Toc500162234)

[**Kernel** 10](#_Toc500162235)

[**Program execution** 10](#_Toc500162236)

[**Interrupts** 11](#_Toc500162237)

[**Modes** 11](#_Toc500162238)

[**Memory management** 12](#_Toc500162239)

[**Virtual memory** 13](#_Toc500162240)

[**Multitasking** 14](#_Toc500162241)

[**Disk access and file systems** 15](#_Toc500162242)

[**Device drivers** 16](#_Toc500162243)

[**Networking** 16](#_Toc500162244)

[**Security** 17](#_Toc500162245)

[**User interface** 18](#_Toc500162246)

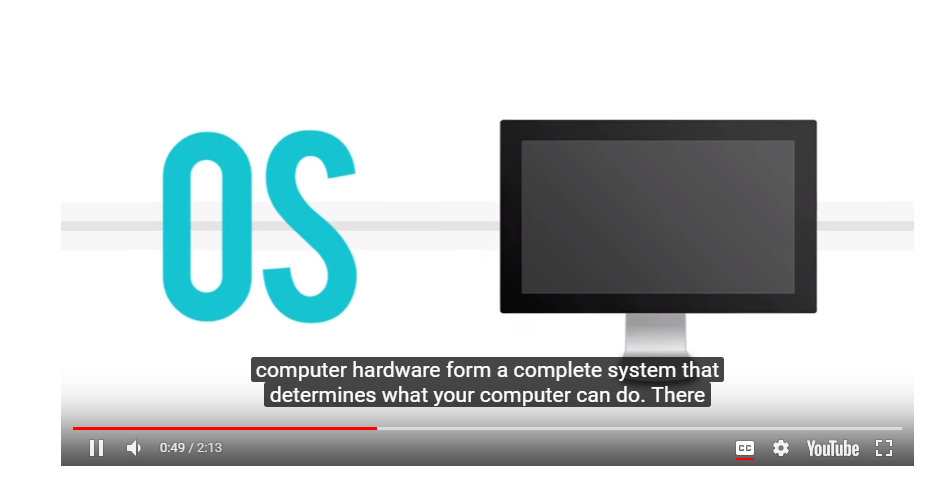
[Graphical user interfaces 19](#_Toc500162247)

[**References** 20](#_Toc500162248)

# **Operating systems**

An **operating system** is the **most important software** that runs on a computer. It manages the computer's **memory** and **processes**, as well as all of its **software** and **hardware**. It also allows you to **communicate** with the computer without knowing how to speak the computer's language. **Without an operating system, a computer is useless**.

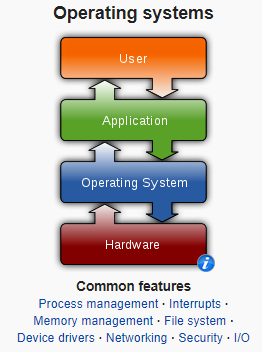
See a video: <https://www.youtube.com/watch?v=pTdSs8kQqSA>



An **operating system** (**OS**) is [system software](https://en.wikipedia.org/wiki/System_software) that manages [computer hardware](https://en.wikipedia.org/wiki/Computer_hardware) and [software](https://en.wikipedia.org/wiki/Computer_software) resources and provides common [services](https://en.wikipedia.org/wiki/Daemon_(computing)) for [computer programs](https://en.wikipedia.org/wiki/Computer_program). All computer programs, excluding [firmware](https://en.wikipedia.org/wiki/Firmware), require an operating system to function.

Operating systems are found on many devices that contain a computer – from [cellular phones](https://en.wikipedia.org/wiki/Cellular_phone) and [video game consoles](https://en.wikipedia.org/wiki/Video_game_console) to [web servers](https://en.wikipedia.org/wiki/Web_server) and [supercomputers](https://en.wikipedia.org/wiki/Supercomputer).

The dominant desktop operating system is [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) with a market share of around 83.3%. [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS) by [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) is in second place (11.2%), and the varieties of [Linux](https://en.wikipedia.org/wiki/Linux) are collectively in third place (1.55%).In the [mobile](https://en.wikipedia.org/wiki/Mobile_operating_system) ([smartphone](https://en.wikipedia.org/wiki/Smartphone) and [tablet](https://en.wikipedia.org/wiki/Tablet_computer) combined) sector, according to third quarter 2016 data, [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) by [Google](https://en.wikipedia.org/wiki/Google) is dominant with 87.5 percent and a growth rate 10.3 percent per year, followed by [iOS](https://en.wikipedia.org/wiki/IOS) by Apple with 12.1 percent and a per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent. [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution) are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.



# **Types of operating systems**

### Single- and multi-tasking

A single-tasking system can only run one program at a time, while a [multi-tasking](https://en.wikipedia.org/wiki/Multi-tasking) operating system allows more than one program to be running in concurrency. This is achieved by [time-sharing](https://en.wikipedia.org/wiki/Time-sharing), dividing the available processor time between multiple processes that are each interrupted repeatedly in time slices by a task-scheduling subsystem of the operating system. Multi-tasking may be characterized in preemptive and co-operative types. In preemptive multitasking, the operating system slices the [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) time and dedicates a slot to each of the programs. Unix-like operating systems, e.g., Solaris, [Linux](https://en.wikipedia.org/wiki/Linux), as well as [AmigaOS](https://en.wikipedia.org/wiki/AmigaOS" \o "AmigaOS) support preemptive multitasking. Cooperative multitasking is achieved by relying on each process to provide time to the other processes in a defined manner. [16-bit](https://en.wikipedia.org/wiki/16-bit) versions of Microsoft Windows used cooperative multi-tasking. [32-bit](https://en.wikipedia.org/wiki/32-bit) versions of both Windows NT and Win9x, used preemptive multi-tasking.

### Single- and multi-user

Single-user operating systems have no facilities to distinguish users, but may allow multiple programs to run in tandem.[[5]](https://en.wikipedia.org/wiki/Operating_system#cite_note-5) A [multi-user](https://en.wikipedia.org/wiki/Multi-user) operating system extends the basic concept of multi-tasking with facilities that identify processes and resources, such as disk space, belonging to multiple users, and the system permits multiple users to interact with the system at the same time. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources to multiple users.

### Distributed

A [distributed operating system](https://en.wikipedia.org/wiki/Distributed_operating_system) manages a group of distinct computers and makes them appear to be a single computer. The development of networked computers that could be linked and communicate with each other gave rise to distributed computing. Distributed computations are carried out on more than one machine. When computers in a group work in cooperation, they form a distributed system.[[6]](https://en.wikipedia.org/wiki/Operating_system#cite_note-6)

### Templated

In an OS, distributed and [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) context, [templating](https://en.wikipedia.org/wiki/Glossary_of_operating_systems_terms" \o "Glossary of operating systems terms) refers to creating a single [virtual machine image](https://en.wikipedia.org/wiki/Virtual_machine_image) as a guest operating system, then saving it as a tool for multiple running virtual machines. The technique is used both in [virtualization](https://en.wikipedia.org/wiki/Virtualization) and cloud computing management, and is common in large server warehouses.

### Embedded

[Embedded operating systems](https://en.wikipedia.org/wiki/Embedded_operating_system) are designed to be used in [embedded computer systems](https://en.wikipedia.org/wiki/Embedded_system). They are designed to operate on small machines like PDAs with less autonomy. They are able to operate with a limited number of resources. They are very compact and extremely efficient by design. Windows CE and Minix 3 are some examples of embedded operating systems.

### Real-time

A [real-time operating system](https://en.wikipedia.org/wiki/Real-time_operating_system) is an operating system that guarantees to process events or data by a specific moment in time. A real-time operating system may be single- or multi-tasking, but when multitasking, it uses specialized scheduling algorithms so that a deterministic nature of behavior is achieved. An event-driven system switches between tasks based on their priorities or external events while time-sharing operating systems switch tasks based on clock interrupts

# **History of OS**

In the 1940s, the earliest electronic digital systems had no operating systems. Electronic systems of this time were programmed on rows of mechanical switches or by jumper wires on plug boards. These were special-purpose systems that, for example, generated ballistics tables for the military or controlled the printing of payroll checks from data on punched paper cards. After programmable general purpose computers were invented, machine languages (consisting of strings of the binary digits 0 and 1 on punched paper tape) were introduced that sped up the programming process

[](https://en.wikipedia.org/wiki/File:IBM360-65-1.corestore.jpg)

[OS/360](https://en.wikipedia.org/wiki/OS/360) was used on most IBM mainframe computers beginning in 1966, including computers used by the [Apollo program](https://en.wikipedia.org/wiki/Apollo_program).

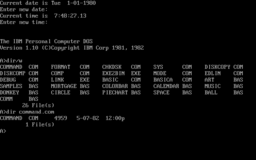
In the early 1950s, a computer could execute only one program at a time. Each user had sole use of the computer for a limited period of time and would arrive at a scheduled time with program and data on punched paper cards or punched tape. The program would be loaded into the machine, and the machine would be set to work until the program completed or crashed. Programs could generally be debugged via a front panel using toggle switches and panel lights. It is said that [Alan Turing](https://en.wikipedia.org/wiki/Alan_Turing) was a master of this on the early [Manchester Mark 1](https://en.wikipedia.org/wiki/Manchester_Mark_1) machine, and he was already deriving the primitive conception of an operating system from the principles of the [universal Turing machine](https://en.wikipedia.org/wiki/Universal_Turing_machine).

## **Mainframes**

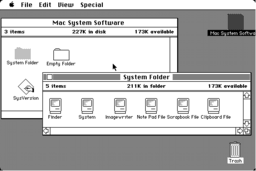
Through the 1950s, many major features were pioneered in the field of operating systems on [mainframe computers](https://en.wikipedia.org/wiki/Mainframe_computer), including [batch processing](https://en.wikipedia.org/wiki/Batch_processing), input/output [interrupt](https://en.wikipedia.org/wiki/Interrupt), [buffering](https://en.wikipedia.org/wiki/Data_buffer), [multitasking](https://en.wikipedia.org/wiki/Computer_multitasking), [spooling](https://en.wikipedia.org/wiki/Spooling), [runtime libraries](https://en.wikipedia.org/wiki/Runtime_library), [link-loading](https://en.wikipedia.org/wiki/Linker_(computing)), and programs for [sorting records](https://en.wikipedia.org/wiki/Sorting_algorithm) in files. These features were included or not included in application software at the option of application programmers, rather than in a separate operating system used by all applications. In 1959, the [SHARE Operating System](https://en.wikipedia.org/wiki/SHARE_Operating_System) was released as an integrated utility for the [IBM 704](https://en.wikipedia.org/wiki/IBM_704), and later in the [709](https://en.wikipedia.org/wiki/IBM_709) and [7090](https://en.wikipedia.org/wiki/IBM_7090) mainframes, although it was quickly supplanted by [IBSYS](https://en.wikipedia.org/wiki/IBM_7090/94_IBSYS)/IBJOB on the 709, 7090 and 7094.

During the 1960s, IBM's [OS/360](https://en.wikipedia.org/wiki/OS/360) introduced the concept of a single OS spanning an entire product line, which was crucial for the success of the System/360 machines. [IBM](https://en.wikipedia.org/wiki/IBM)'s current mainframe operating systems are [distant descendants](https://en.wikipedia.org/wiki/History_of_IBM_mainframe_operating_systems) of this original system and applications written for OS/360 can still be run on modern machines

## **Microcomputers**

[](https://en.wikipedia.org/wiki/File:PC_DOS_1.10_screenshot.png)

PC DOS was an early personal computer OS that featured a command line interface.

[](https://en.wikipedia.org/wiki/File:Apple_Macintosh_Desktop.png)

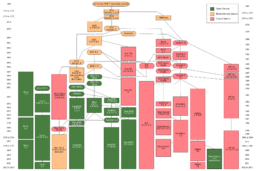
Mac OS by [Apple Computer](https://en.wikipedia.org/wiki/Apple_Computer) became the first widespread OS to feature a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface). Many of its features such as windows and icons would later become commonplace in GUIs.

The first [microcomputers](https://en.wikipedia.org/wiki/Microcomputer) did not have the capacity or need for the elaborate operating systems that had been developed for mainframes and minis; minimalistic operating systems were developed, often loaded from [ROM](https://en.wikipedia.org/wiki/Read-only_memory) and known as [*monitors*](https://en.wikipedia.org/wiki/Resident_monitor). One notable early [disk operating system](https://en.wikipedia.org/wiki/Disk_operating_system) was [CP/M](https://en.wikipedia.org/wiki/CP/M), which was supported on many early microcomputers and was closely imitated by [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s [MS-DOS](https://en.wikipedia.org/wiki/MS-DOS), which became widely popular as the operating system chosen for the [IBM PC](https://en.wikipedia.org/wiki/IBM_PC) (IBM's version of it was called IBM DOS or [PC DOS](https://en.wikipedia.org/wiki/PC_DOS)). In the 1980s, Apple Computer Inc. (now [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.)) abandoned its popular [Apple II](https://en.wikipedia.org/wiki/Apple_II) series of microcomputers to introduce the [Apple Macintosh](https://en.wikipedia.org/wiki/Apple_Macintosh) computer with an innovative [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) to the [Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS).

The introduction of the [Intel 80386](https://en.wikipedia.org/wiki/Intel_80386) CPU chip in October 1985,[[11]](https://en.wikipedia.org/wiki/Operating_system#cite_note-11) with [32-bit](https://en.wikipedia.org/wiki/32-bit) architecture and [paging](https://en.wikipedia.org/wiki/Paging) capabilities, provided personal computers with the ability to run [multitasking](https://en.wikipedia.org/wiki/Computer_multitasking) operating systems like those of earlier [minicomputers](https://en.wikipedia.org/wiki/Minicomputer) and [mainframes](https://en.wikipedia.org/wiki/Mainframe_computer). Microsoft responded to this progress by hiring [Dave Cutler](https://en.wikipedia.org/wiki/Dave_Cutler), who had developed the [VMS](https://en.wikipedia.org/wiki/OpenVMS) operating system for [Digital Equipment Corporation](https://en.wikipedia.org/wiki/Digital_Equipment_Corporation). He would lead the development of the [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) operating system, which continues to serve as the basis for Microsoft's operating systems line. [Steve Jobs](https://en.wikipedia.org/wiki/Steve_Jobs), a co-founder of [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.), started [NeXT](https://en.wikipedia.org/wiki/NeXT) Computer Inc., which developed the [NEXTSTEP](https://en.wikipedia.org/wiki/NEXTSTEP) operating system. NEXTSTEP would later be acquired by [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) and used, along with code from [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) as the core of [Mac OS X](https://en.wikipedia.org/wiki/MacOS) (macOS after latest name change).

The [GNU Project](https://en.wikipedia.org/wiki/GNU_Project) was started by activist and programmer [Richard Stallman](https://en.wikipedia.org/wiki/Richard_Stallman) with the goal of creating a complete [free software](https://en.wikipedia.org/wiki/Free_software)replacement to the proprietary [UNIX](https://en.wikipedia.org/wiki/UNIX) operating system.

## **Unix and Unix-like operating systems**

[](https://en.wikipedia.org/wiki/File:Unix_history-simple.svg)

Evolution of [Unix](https://en.wikipedia.org/wiki/Unix) systems

Unix was originally written in [assembly language](https://en.wikipedia.org/wiki/Assembly_language).[[12]](https://en.wikipedia.org/wiki/Operating_system#cite_note-12) [Ken Thompson](https://en.wikipedia.org/wiki/Ken_Thompson) wrote [B](https://en.wikipedia.org/wiki/B_(programming_language)), mainly based on [BCPL](https://en.wikipedia.org/wiki/BCPL), based on his experience in the [MULTICS](https://en.wikipedia.org/wiki/MULTICS) project. B was replaced by [C](https://en.wikipedia.org/wiki/C_(programming_language)), and Unix, rewritten in C, developed into a large, complex family of inter-related operating systems which have been influential in every modern operating system (see [History](https://en.wikipedia.org/wiki/History_of_operating_systems)).

The [*Unix-like*](https://en.wikipedia.org/wiki/Unix-like) family is a diverse group of operating systems, with several major sub-categories including [System V](https://en.wikipedia.org/wiki/System_V), [BSD](https://en.wikipedia.org/wiki/Berkeley_Software_Distribution), and [Linux](https://en.wikipedia.org/wiki/Linux). The name "[UNIX](https://en.wikipedia.org/wiki/Unix)" is a trademark of [The Open Group](https://en.wikipedia.org/wiki/The_Open_Group) which licenses it for use with any operating system that has been shown to conform to their definitions. "UNIX-like" is commonly used to refer to the large set of operating systems which resemble the original UNIX.

Unix-like systems run on a wide variety of [computer architectures](https://en.wikipedia.org/wiki/Computer_architecture). They are used heavily for [servers](https://en.wikipedia.org/wiki/Server_(computing)) in business, as well as [workstations](https://en.wikipedia.org/wiki/Workstation) in academic and engineering environments. [Free](https://en.wikipedia.org/wiki/Free_software) UNIX variants, such as [Linux](https://en.wikipedia.org/wiki/Linux) and [BSD](https://en.wikipedia.org/wiki/Berkeley_Software_Distribution), are popular in these areas.

Four operating systems are certified by [The Open Group](https://en.wikipedia.org/wiki/The_Open_Group) (holder of the Unix trademark) as Unix. HP's [HP-UX](https://en.wikipedia.org/wiki/HP-UX) and IBM's [AIX](https://en.wikipedia.org/wiki/AIX_operating_system) are both descendants of the original System V Unix and are designed to run only on their respective vendor's hardware. In contrast, [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)'s [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) can run on multiple types of hardware, including [x86](https://en.wikipedia.org/wiki/X86) and [Sparc](https://en.wikipedia.org/wiki/Sparc" \o "Sparc) servers, and PCs. Apple's [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), a replacement for Apple's earlier (non-Unix) Mac OS, is a [hybrid kernel](https://en.wikipedia.org/wiki/Hybrid_kernel)-based BSD variant derived from [NeXTSTEP](https://en.wikipedia.org/wiki/NeXTSTEP" \o "NeXTSTEP), [Mach](https://en.wikipedia.org/wiki/Mach_(kernel)), and [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD).

## **BSD and its descendants**

[](https://en.wikipedia.org/wiki/File:First_Web_Server.jpg)

The [first server](https://en.wikipedia.org/wiki/CERN_httpd) for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web)ran on NeXTSTEP, based on BSD.

A subgroup of the Unix family is the [Berkeley Software Distribution](https://en.wikipedia.org/wiki/Berkeley_Software_Distribution) family, which includes [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD), [NetBSD](https://en.wikipedia.org/wiki/NetBSD" \o "NetBSD), and [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD" \o "OpenBSD). These operating systems are most commonly found on [webservers](https://en.wikipedia.org/wiki/Webserver), although they can also function as a personal computer OS. The Internet owes much of its existence to BSD, as many of the protocols now commonly used by computers to connect, send and receive data over a network were widely implemented and refined in BSD. The [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) was also first demonstrated on a number of computers running an OS based on BSD called [NeXTSTEP](https://en.wikipedia.org/wiki/NeXTSTEP" \o "NeXTSTEP).

In 1974, [University of California, Berkeley](https://en.wikipedia.org/wiki/University_of_California,_Berkeley) installed its first Unix system. Over time, students and staff in the computer science department there began adding new programs to make things easier, such as text editors. When Berkeley received new [VAX](https://en.wikipedia.org/wiki/VAX" \o "VAX)computers in 1978 with Unix installed, the school's undergraduates modified Unix even more in order to take advantage of the computer's hardware possibilities. The [Defense Advanced Research Projects Agency](https://en.wikipedia.org/wiki/Defense_Advanced_Research_Projects_Agency) of the US [Department of Defense](https://en.wikipedia.org/wiki/United_States_Department_of_Defense) took interest, and decided to fund the project. Many schools, corporations, and government organizations took notice and started to use Berkeley's version of Unix instead of the official one distributed by AT&T.

[Steve Jobs](https://en.wikipedia.org/wiki/Steve_Jobs), upon leaving Apple Inc. in 1985, formed [NeXT Inc.](https://en.wikipedia.org/wiki/NeXT), a company that manufactured high-end computers running on a variation of BSD called [NeXTSTEP](https://en.wikipedia.org/wiki/NeXTSTEP" \o "NeXTSTEP). One of these computers was used by [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee) as the first webserver to create the World Wide Web.

Developers like [Keith Bostic](https://en.wikipedia.org/wiki/Keith_Bostic) encouraged the project to replace any non-free code that originated with Bell Labs. Once this was done, however, AT&T sued. After two years of legal disputes, the BSD project spawned a number of free derivatives, such as [NetBSD](https://en.wikipedia.org/wiki/NetBSD" \o "NetBSD) and [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) (both in 1993), and [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD" \o "OpenBSD) (from NetBSD in 1995).

## **macOS**

**macOS** (formerly "Mac OS X" and later "OS X") is a line of [open core](https://en.wikipedia.org/wiki/Open_core) graphical operating systems developed, marketed, and sold by [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.), the latest of which is pre-loaded on all currently shipping [Macintosh](https://en.wikipedia.org/wiki/Macintosh) computers. macOS is the successor to the original [classic Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS), which had been Apple's primary operating system since 1984. Unlike its predecessor, macOS is a [UNIX](https://en.wikipedia.org/wiki/UNIX) operating system built on technology that had been developed at [NeXT](https://en.wikipedia.org/wiki/NeXT) through the second half of the 1980s and up until Apple purchased the company in early 1997. The operating system was first released in 1999 as [Mac OS X Server 1.0](https://en.wikipedia.org/wiki/Mac_OS_X_Server_1.0), followed in March 2001 by a client version ([Mac OS X v10.0 "Cheetah"](https://en.wikipedia.org/wiki/Mac_OS_X_v10.0)). Since then, six more distinct "client" and "[server](https://en.wikipedia.org/wiki/MacOS_Server)" editions of macOS have been released, until the two were merged in [OS X 10.7 "Lion"](https://en.wikipedia.org/wiki/Mac_OS_X_Lion).

Prior to its merging with macOS, the server edition – [macOS Server](https://en.wikipedia.org/wiki/MacOS_Server" \o "MacOS Server) – was [architecturally](https://en.wikipedia.org/wiki/Software_architecture) identical to its desktop counterpart and usually ran on Apple's line of Macintosh [server](https://en.wikipedia.org/wiki/Server_(computing)" \o "Server (computing))hardware. macOS Server included work group management and administration software tools that provide simplified access to key [network services](https://en.wikipedia.org/wiki/Network_service), including a [mail transfer agent](https://en.wikipedia.org/wiki/Message_transfer_agent), a [Samba server](https://en.wikipedia.org/wiki/Samba_(software)), an [LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol) server, a [domain name server](https://en.wikipedia.org/wiki/Domain_Name_System), and others. With [Mac OS X v10.7 Lion](https://en.wikipedia.org/wiki/Mac_OS_X_Lion), all server aspects of Mac OS X Server have been integrated into the client version and the product re-branded as "OS X" (dropping "Mac" from the name). The server tools are now offered as an application.[[13]](https://en.wikipedia.org/wiki/Operating_system#cite_note-13)

## **Linux**

[](https://en.wikipedia.org/wiki/File:Ubuntu_16.10_English.png)

[Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), desktop [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution)

The Linux kernel originated in 1991, as a project of [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds), while a university student in Finland. He posted information about his project on a newsgroup for computer students and programmers, and received support and assistance from volunteers who succeeded in creating a complete and functional kernel.

[Linux](https://en.wikipedia.org/wiki/Linux) is [Unix-like](https://en.wikipedia.org/wiki/Unix-like), but was developed without any Unix code, unlike BSD and its variants. Because of its open license model, the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) code is available for study and modification, which resulted in its use on a wide range of computing machinery from supercomputers to smart-watches. Although estimates suggest that Linux is [used on](https://en.wikipedia.org/wiki/Usage_share_of_operating_systems) only 1.82% of all "desktop" (or laptop) PCs,[[14]](https://en.wikipedia.org/wiki/Operating_system#cite_note-StatCounter-14) it has been widely adopted for use in servers[[15]](https://en.wikipedia.org/wiki/Operating_system#cite_note-15) and embedded systems[[16]](https://en.wikipedia.org/wiki/Operating_system#cite_note-16) such as cell phones. Linux has superseded Unix on many platforms and is used on most supercomputers including the top 385.[[17]](https://en.wikipedia.org/wiki/Operating_system#cite_note-top500-list-17) Many of the same computers are also on [Green500](https://en.wikipedia.org/wiki/Green500) (but in different order), and Linux runs on the top 10. Linux is also commonly used on other small energy-efficient computers, such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [smartwatches](https://en.wikipedia.org/wiki/Smartwatch). The Linux kernel is used in some popular distributions, such as [Red Hat](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux), [Debian](https://en.wikipedia.org/wiki/Debian" \o "Debian), [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), [Linux Mint](https://en.wikipedia.org/wiki/Linux_Mint)and [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [Chrome OS](https://en.wikipedia.org/wiki/Chrome_OS), and [Chromium OS](https://en.wikipedia.org/wiki/Chromium_OS).

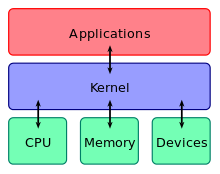
## **Microsoft Windows**

Microsoft Windows is a family of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) operating systems designed by [Microsoft Corporation](https://en.wikipedia.org/wiki/Microsoft) and primarily targeted to Intel architecture based computers, with an estimated 88.9 percent total usage share on Web connected computers.[[14]](https://en.wikipedia.org/wiki/Operating_system#cite_note-StatCounter-14)[[18]](https://en.wikipedia.org/wiki/Operating_system#cite_note-18)[[19]](https://en.wikipedia.org/wiki/Operating_system#cite_note-w3cstats-19)[[20]](https://en.wikipedia.org/wiki/Operating_system#cite_note-20) The latest version is [Windows 10](https://en.wikipedia.org/wiki/Windows_10).

In 2011, Windows 7 overtook Windows XP as most common version in use. Microsoft Windows was first released in 1985, as an [operating environment](https://en.wikipedia.org/wiki/Operating_environment) running on top of [MS-DOS](https://en.wikipedia.org/wiki/MS-DOS), which was the standard operating system shipped on most Intel architecture personal computers at the time. In 1995, [Windows 95](https://en.wikipedia.org/wiki/Windows_95) was released which only used MS-DOS as a bootstrap. For backwards compatibility, Win9x could run real-mode MS-DOS[[24]](https://en.wikipedia.org/wiki/Operating_system" \l "cite_note-24)[[25]](https://en.wikipedia.org/wiki/Operating_system#cite_note-25) and 16-bit [Windows 3.x](https://en.wikipedia.org/wiki/Windows_3.x)[[26]](https://en.wikipedia.org/wiki/Operating_system#cite_note-26) drivers. [Windows ME](https://en.wikipedia.org/wiki/Windows_ME), released in 2000, was the last version in the Win9x family. Later versions have all been based on the [Windows NT](https://en.wikipedia.org/wiki/Windows_NT)[kernel](https://en.wikipedia.org/wiki/Kernel_(computing)). Current client versions of Windows run on [IA-32](https://en.wikipedia.org/wiki/IA-32), [x86-64](https://en.wikipedia.org/wiki/X86-64) and 32-bit [ARM](https://en.wikipedia.org/wiki/ARMv7) [microprocessors](https://en.wikipedia.org/wiki/Microprocessor).[[27]](https://en.wikipedia.org/wiki/Operating_system#cite_note-27) In addition [Itanium](https://en.wikipedia.org/wiki/Itanium) is still supported in older server version [Windows Server 2008 R2](https://en.wikipedia.org/wiki/Windows_Server_2008_R2). In the past, Windows NT supported additional architectures.

# **Components of OS**

## **Kernel**

[](https://en.wikipedia.org/wiki/File:Kernel_Layout.svg)

A kernel connects the application software to the hardware of a computer.

With the aid of the [firmware](https://en.wikipedia.org/wiki/Firmware) and [device drivers](https://en.wikipedia.org/wiki/Device_driver), the kernel provides the most basic level of control over all of the computer's hardware devices. It manages memory access for programs in the [RAM](https://en.wikipedia.org/wiki/Random-access_memory), it determines which programs get access to which hardware resources, it sets up or resets the CPU's operating states for optimal operation at all times, and it organizes the data for long-term [non-volatile storage](https://en.wikipedia.org/wiki/Non-volatile_storage) with [file systems](https://en.wikipedia.org/wiki/File_system) on such media as disks, tapes, flash memory, etc.

## **Program execution**

*Main article:*[*Process (computing)*](https://en.wikipedia.org/wiki/Process_(computing))

The operating system provides an interface between an application program and the computer hardware, so that an application program can interact with the hardware only by obeying rules and procedures programmed into the operating system. The operating system is also a set of services which simplify development and execution of application programs. Executing an application program involves the creation of a process by the operating system [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) which assigns memory space and other resources, establishes a priority for the process in multi-tasking systems, loads program binary code into memory, and initiates execution of the application program which then interacts with the user and with hardware devices.

## **Interrupts**

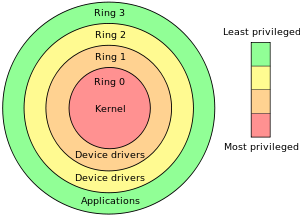
[Interrupts](https://en.wikipedia.org/wiki/Interrupt) are central to operating systems, as they provide an efficient way for the operating system to interact with and react to its environment. The alternative – having the operating system "watch" the various sources of input for events (polling) that require action – can be found in older systems with very small [stacks](https://en.wikipedia.org/wiki/Call_stack) (50 or 60 bytes) but is unusual in modern systems with large stacks. [Interrupt](https://en.wikipedia.org/wiki/Interrupt)-based programming is directly supported by most modern CPUs. Interrupts provide a computer with a way of automatically saving local register contexts, and running specific code in response to events. Even very basic computers support hardware interrupts, and allow the programmer to specify code which may be run when that event takes place.

When an interrupt is received, the computer's hardware automatically suspends whatever program is currently running, saves its status, and runs computer code previously associated with the interrupt; this is analogous to placing a bookmark in a book in response to a phone call. In modern operating systems, interrupts are handled by the operating system's [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)). Interrupts may come from either the computer's hardware or the running program.

When a hardware device triggers an interrupt, the operating system's kernel decides how to deal with this event, generally by running some processing code. The amount of code being run depends on the priority of the interrupt (for example: a person usually responds to a smoke detector alarm before answering the phone). The processing of hardware interrupts is a task that is usually delegated to software called a [device driver](https://en.wikipedia.org/wiki/Device_drivers), which may be part of the operating system's kernel, part of another program, or both. Device drivers may then relay information to a running program by various means.

A program may also trigger an interrupt to the operating system. If a program wishes to access hardware, for example, it may interrupt the operating system's kernel, which causes control to be passed back to the kernel. The kernel then processes the request. If a program wishes additional resources (or wishes to shed resources) such as memory, it triggers an interrupt to get the kernel's attention.

## **Modes**

[](https://en.wikipedia.org/wiki/File:Priv_rings.svg)

Privilege rings for the [x86](https://en.wikipedia.org/wiki/X86) microprocessor architecture available in [protected mode](https://en.wikipedia.org/wiki/Protected_mode). Operating systems determine which processes run in each mode.

Modern [microprocessors](https://en.wikipedia.org/wiki/Microprocessor) (CPU or MPU) support multiple modes of operation. CPUs with this capability offer at least two modes: [user mode](https://en.wikipedia.org/wiki/User_mode) and [supervisor mode](https://en.wikipedia.org/wiki/Supervisor_mode). In general terms, supervisor mode operation allows unrestricted access to all machine resources, including all MPU instructions. User mode operation sets limits on instruction use and typically disallows direct access to machine resources. CPUs might have other modes similar to user mode as well, such as the virtual modes in order to emulate older processor types, such as 16-bit processors on a 32-bit one, or 32-bit processors on a [64-bit](https://en.wikipedia.org/wiki/64-bit_computing) one.

At power-on or reset, the system begins in supervisor mode. Once an operating system [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) has been loaded and started, the boundary between user mode and supervisor mode (also known as kernel mode) can be established.

Supervisor mode is used by the kernel for low level tasks that need unrestricted access to hardware, such as controlling how memory is accessed, and communicating with devices such as disk drives and video display devices. User mode, in contrast, is used for almost everything else. Application programs, such as word processors and database managers, operate within user mode, and can only access machine resources by turning control over to the kernel, a process which causes a switch to supervisor mode. Typically, the transfer of control to the kernel is achieved by executing a [software interrupt](https://en.wikipedia.org/wiki/Software_interrupt) instruction, such as the Motorola 68000 TRAP instruction. The software interrupt causes the microprocessor to switch from user mode to supervisor mode and begin executing code that allows the kernel to take control.

In user mode, programs usually have access to a restricted set of microprocessor instructions, and generally cannot execute any instructions that could potentially cause disruption to the system's operation. In supervisor mode, instruction execution restrictions are typically removed, allowing the kernel unrestricted access to all machine resources.

The term "user mode resource" generally refers to one or more CPU registers, which contain information that the running program isn't allowed to alter. Attempts to alter these resources generally causes a switch to supervisor mode, where the operating system can deal with the illegal operation the program was attempting, for example, by forceably terminating ("killing") the program).

## **Memory management**

Among other things, a multiprogramming operating system [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) must be responsible for managing all system memory which is currently in use by programs. This ensures that a program does not interfere with memory already in use by another program. Since programs time share, each program must have independent access to memory.

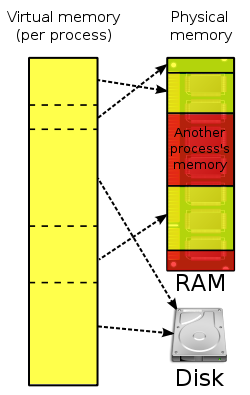
Cooperative memory management, used by many early operating systems, assumes that all programs make voluntary use of the [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science))'s memory manager, and do not exceed their allocated memory. This system of memory management is almost never seen any more, since programs often contain bugs which can cause them to exceed their allocated memory. If a program fails, it may cause memory used by one or more other programs to be affected or overwritten. Malicious programs or viruses may purposefully alter another program's memory, or may affect the operation of the operating system itself. With cooperative memory management, it takes only one misbehaved program to crash the system.

[Memory protection](https://en.wikipedia.org/wiki/Memory_protection) enables the [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) to limit a process' access to the computer's memory. Various methods of memory protection exist, including [memory segmentation](https://en.wikipedia.org/wiki/Memory_segmentation) and [paging](https://en.wikipedia.org/wiki/Paging). All methods require some level of hardware support (such as the [80286](https://en.wikipedia.org/wiki/80286) MMU), which doesn't exist in all computers.

In both segmentation and paging, certain [protected mode](https://en.wikipedia.org/wiki/Protected_mode) registers specify to the CPU what memory address it should allow a running program to access. Attempts to access other addresses trigger an interrupt which cause the CPU to re-enter [supervisor mode](https://en.wikipedia.org/wiki/Supervisor_mode), placing the [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) in charge. This is called a [segmentation violation](https://en.wikipedia.org/wiki/Segmentation_violation) or Seg-V for short, and since it is both difficult to assign a meaningful result to such an operation, and because it is usually a sign of a misbehaving program, the [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) generally resorts to terminating the offending program, and reports the error.

Windows versions 3.1 through ME had some level of memory protection, but programs could easily circumvent the need to use it. A [general protection fault](https://en.wikipedia.org/wiki/General_protection_fault) would be produced, indicating a segmentation violation had occurred; however, the system would often crash anyway.

## **Virtual memory**

[](https://en.wikipedia.org/wiki/File:Virtual_memory.svg)

Many operating systems can "trick" programs into using memory scattered around the hard disk and RAM as if it is one continuous chunk of memory, called virtual memory.

The use of virtual memory addressing (such as paging or segmentation) means that the kernel can choose what memory each program may use at any given time, allowing the operating system to use the same memory locations for multiple tasks.

If a program tries to access memory that isn't in its current range of accessible memory, but nonetheless has been allocated to it, the kernel is interrupted in the same way as it would if the program were to exceed its allocated memory. (See section on memory management.) Under UNIX this kind of interrupt is referred to as a [page fault](https://en.wikipedia.org/wiki/Page_fault).

When the kernel detects a page fault it generally adjusts the virtual memory range of the program which triggered it, granting it access to the memory requested. This gives the kernel discretionary power over where a particular application's memory is stored, or even whether or not it has actually been allocated yet.

In modern operating systems, memory which is accessed less frequently can be temporarily stored on disk or other media to make that space available for use by other programs. This is called [swapping](https://en.wikipedia.org/wiki/Paging), as an area of memory can be used by multiple programs, and what that memory area contains can be swapped or exchanged on demand.

"Virtual memory" provides the programmer or the user with the perception that there is a much larger amount of RAM in the computer than is really there.

## **Multitasking**

[Multitasking](https://en.wikipedia.org/wiki/Computer_multitasking) refers to the running of multiple independent computer programs on the same computer; giving the appearance that it is performing the tasks at the same time. Since most computers can do at most one or two things at one time, this is generally done via time-sharing, which means that each program uses a share of the computer's time to execute.

An operating system [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)) contains a [scheduling](https://en.wikipedia.org/wiki/Scheduling_(computing)) program which determines how much time each process spends executing, and in which order execution control should be passed to programs. Control is passed to a process by the kernel, which allows the program access to the [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) and memory. Later, control is returned to the kernel through some mechanism, so that another program may be allowed to use the CPU. This so-called passing of control between the kernel and applications is called a [context switch](https://en.wikipedia.org/wiki/Context_switch).

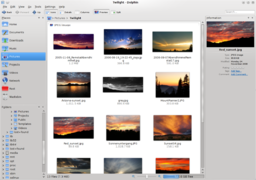
An early model which governed the allocation of time to programs was called [cooperative multitasking](https://en.wikipedia.org/wiki/Cooperative_multitasking). In this model, when control is passed to a program by the kernel, it may execute for as long as it wants before explicitly returning control to the kernel. This means that a malicious or malfunctioning program may not only prevent any other programs from using the CPU, but it can hang the entire system if it enters an [infinite loop](https://en.wikipedia.org/wiki/Infinite_loop).

Modern operating systems extend the concepts of application preemption to device drivers and kernel code, so that the operating system has preemptive control over internal run-times as well.

The philosophy governing [preemptive multitasking](https://en.wikipedia.org/wiki/Preemptive_multitasking) is that of ensuring that all programs are given regular time on the CPU. This implies that all programs must be limited in how much time they are allowed to spend on the CPU without being interrupted. To accomplish this, modern operating system kernels make use of a timed interrupt. A [protected mode](https://en.wikipedia.org/wiki/Protected_mode) timer is set by the kernel which triggers a return to supervisor mode after the specified time has elapsed. (See above sections on Interrupts and Dual Mode Operation.)

On many single user operating systems cooperative multitasking is perfectly adequate, as home computers generally run a small number of well tested programs. The [AmigaOS](https://en.wikipedia.org/wiki/AmigaOS" \o "AmigaOS) is an exception, having preemptive multitasking from its very first version. [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) was the first version of [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) which enforced preemptive multitasking, but it didn't reach the home user market until [Windows XP](https://en.wikipedia.org/wiki/Windows_XP) (since [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) was targeted at professionals).

## **Disk access and file systems**

[](https://en.wikipedia.org/wiki/File:Dolphin_FileManager.png)

File systems allow users and programs to organize and sort files on a computer, often through the use of directories (or "folders").

Access to data stored on disks is a central feature of all operating systems. Computers store data on [disks](https://en.wikipedia.org/wiki/Hard_disk_drive) using [files](https://en.wikipedia.org/wiki/Computer_file), which are structured in specific ways in order to allow for faster access, higher reliability, and to make better use of the drive's available space. The specific way in which files are stored on a disk is called a [file system](https://en.wikipedia.org/wiki/File_system), and enables files to have names and attributes. It also allows them to be stored in a hierarchy of directories or folders arranged in a [directory tree](https://en.wikipedia.org/wiki/Directory_(computing)).

Early operating systems generally supported a single type of disk drive and only one kind of file system. Early file systems were limited in their capacity, speed, and in the kinds of file names and directory structures they could use. These limitations often reflected limitations in the operating systems they were designed for, making it very difficult for an operating system to support more than one file system.

While many simpler operating systems support a limited range of options for accessing storage systems, operating systems like [UNIX](https://en.wikipedia.org/wiki/Unix" \o "Unix)and [Linux](https://en.wikipedia.org/wiki/Linux) support a technology known as a [virtual file system](https://en.wikipedia.org/wiki/Virtual_file_system) or VFS. An operating system such as UNIX supports a wide array of storage devices, regardless of their design or [file systems](https://en.wikipedia.org/wiki/File_system), allowing them to be accessed through a common [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API). This makes it unnecessary for programs to have any knowledge about the device they are accessing. A VFS allows the operating system to provide programs with access to an unlimited number of devices with an infinite variety of file systems installed on them, through the use of specific [device drivers](https://en.wikipedia.org/wiki/Device_driver) and file system drivers.

A connected [storage device](https://en.wikipedia.org/wiki/Data_storage), such as a [hard drive](https://en.wikipedia.org/wiki/Hard_disk_drive), is accessed through a [device driver](https://en.wikipedia.org/wiki/Device_driver). The device driver understands the specific language of the drive and is able to translate that language into a standard language used by the operating system to access all disk drives. On UNIX, this is the language of [block devices](https://en.wikipedia.org/wiki/Block_device).

When the kernel has an appropriate device driver in place, it can then access the contents of the disk drive in raw format, which may contain one or more file systems. A file system driver is used to translate the commands used to access each specific file system into a standard set of commands that the operating system can use to talk to all file systems. Programs can then deal with these file systems on the basis of filenames, and directories/folders, contained within a hierarchical structure. They can create, delete, open, and close files, as well as gather various information about them, including access permissions, size, free space, and creation and modification dates.

Various differences between file systems make supporting all file systems difficult. Allowed characters in file names, [case sensitivity](https://en.wikipedia.org/wiki/Case_sensitivity), and the presence of various kinds of [file attributes](https://en.wikipedia.org/wiki/File_attribute) makes the implementation of a single interface for every file system a daunting task. Operating systems tend to recommend using (and so support natively) file systems specifically designed for them; for example, [NTFS](https://en.wikipedia.org/wiki/NTFS) in Windows and [ext3](https://en.wikipedia.org/wiki/Ext3) and [ReiserFS](https://en.wikipedia.org/wiki/ReiserFS" \o "ReiserFS) in Linux. However, in practice, third party drivers are usually available to give support for the most widely used file systems in most general-purpose operating systems (for example, NTFS is available in Linux through [NTFS-3g](https://en.wikipedia.org/wiki/NTFS-3G), and ext2/3 and ReiserFS are available in Windows through third-party software).

Support for file systems is highly varied among modern operating systems, although there are several common file systems which almost all operating systems include support and drivers for. Operating systems vary on file system support and on the disk formats they may be installed on. Under Windows, each file system is usually limited in application to certain media; for example, CDs must use [ISO 9660](https://en.wikipedia.org/wiki/ISO_9660) or [UDF](https://en.wikipedia.org/wiki/Universal_Disk_Format), and as of [Windows Vista](https://en.wikipedia.org/wiki/Windows_Vista), NTFS is the only file system which the operating system can be installed on. It is possible to install Linux onto many types of file systems. Unlike other operating systems, Linux and UNIX allow any file system to be used regardless of the media it is stored in, whether it is a hard drive, a disc (CD, DVD...), a USB flash drive, or even contained within a file located on another file system.

## **Device drivers**

A [device driver](https://en.wikipedia.org/wiki/Device_driver) is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through the specific computer bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the operating system and software applications. It is a specialized hardware-dependent computer program which is also operating system specific that enables another program, typically an operating system or applications software package or computer program running under the operating system kernel, to interact transparently with a hardware device, and usually provides the requisite interrupt handling necessary for any necessary asynchronous time-dependent hardware interfacing needs.

## **Networking**

Currently most operating systems support a variety of networking protocols, hardware, and applications for using them. This means that computers running dissimilar operating systems can participate in a common [network](https://en.wikipedia.org/wiki/Computer_network) for sharing resources such as [computing](https://en.wikipedia.org/wiki/Remote_procedure_call), files, printers, and scanners using either wired or wireless connections. Networks can essentially allow a computer's operating system to access the resources of a remote computer to support the same functions as it could if those resources were connected directly to the local computer. This includes everything from simple communication, to using networked file systems or even sharing another computer's graphics or sound hardware. Some network services allow the resources of a computer to be accessed transparently, such as [SSH](https://en.wikipedia.org/wiki/Secure_Shell) which allows networked users direct access to a computer's command line interface.

Client/server networking allows a program on a computer, called a client, to connect via a network to another computer, called a server. Servers offer (or host) various services to other network computers and users. These services are usually provided through ports or numbered access points beyond the server's [IP address](https://en.wikipedia.org/wiki/IP_address). Each port number is usually associated with a maximum of one running program, which is responsible for handling requests to that port. A daemon, being a user program, can in turn access the local hardware resources of that computer by passing requests to the operating system kernel.

Many operating systems support one or more vendor-specific or open networking protocols as well, for example, [SNA](https://en.wikipedia.org/wiki/Systems_Network_Architecture) on [IBM](https://en.wikipedia.org/wiki/IBM) systems, [DECnet](https://en.wikipedia.org/wiki/DECnet" \o "DECnet) on systems from [Digital Equipment Corporation](https://en.wikipedia.org/wiki/Digital_Equipment_Corporation), and Microsoft-specific protocols ([SMB](https://en.wikipedia.org/wiki/Server_message_block)) on Windows. Specific protocols for specific tasks may also be supported such as [NFS](https://en.wikipedia.org/wiki/Network_File_System) for file access. Protocols like [ESound](https://en.wikipedia.org/wiki/ESound" \o "ESound), or esd can be easily extended over the network to provide sound from local applications, on a remote system's sound hardware.

## **Security**

A computer being secure depends on a number of technologies working properly. A modern operating system provides access to a number of resources, which are available to software running on the system, and to external devices like networks via the kernel.

The operating system must be capable of distinguishing between requests which should be allowed to be processed, and others which should not be processed. While some systems may simply distinguish between "privileged" and "non-privileged", systems commonly have a form of requester *identity*, such as a user name. To establish identity there may be a process of *authentication*. Often a username must be quoted, and each username may have a password. Other methods of authentication, such as magnetic cards or biometric data, might be used instead. In some cases, especially connections from the network, resources may be accessed with no authentication at all (such as reading files over a network share). Also covered by the concept of requester **identity** is *authorization*; the particular services and resources accessible by the requester once logged into a system are tied to either the requester's user account or to the variously configured groups of users to which the requester belongs.

In addition to the allow or disallow model of security, a system with a high level of security also offers auditing options. These would allow tracking of requests for access to resources (such as, "who has been reading this file?"). Internal security, or security from an already running program is only possible if all possibly harmful requests must be carried out through interrupts to the operating system kernel. If programs can directly access hardware and resources, they cannot be secured.

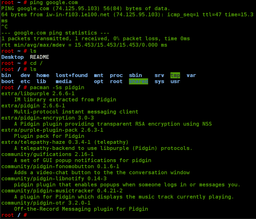
External security involves a request from outside the computer, such as a login at a connected console or some kind of network connection. External requests are often passed through device drivers to the operating system's kernel, where they can be passed onto applications, or carried out directly. Security of operating systems has long been a concern because of highly sensitive data held on computers, both of a commercial and military nature. The United States [Government](https://en.wikipedia.org/wiki/Government_of_the_United_States) [Department of Defense](https://en.wikipedia.org/wiki/United_States_Department_of_Defense) (DoD) created the [*Trusted Computer System Evaluation Criteria*](https://en.wikipedia.org/wiki/Trusted_Computer_System_Evaluation_Criteria) (TCSEC) which is a standard that sets basic requirements for assessing the effectiveness of security. This became of vital importance to operating system makers, because the TCSEC was used to evaluate, classify and select [trusted operating systems](https://en.wikipedia.org/wiki/Trusted_operating_system) being considered for the processing, storage and retrieval of sensitive or [classified information](https://en.wikipedia.org/wiki/Classified_information).

Network services include offerings such as file sharing, print services, email, web sites, and [file transfer protocols](https://en.wikipedia.org/wiki/File_transfer_protocol) (FTP), most of which can have compromised security. At the front line of security are hardware devices known as [firewalls](https://en.wikipedia.org/wiki/Firewall_(networking)) or intrusion detection/prevention systems. At the operating system level, there are a number of software firewalls available, as well as intrusion detection/prevention systems. Most modern operating systems include a software firewall, which is enabled by default. A software firewall can be configured to allow or deny network traffic to or from a service or application running on the operating system. Therefore, one can install and be running an insecure service, such as Telnet or FTP, and not have to be threatened by a security breach because the firewall would deny all traffic trying to connect to the service on that port.

An alternative strategy, and the only [sandbox](https://en.wikipedia.org/wiki/Sandbox_(computer_security)) strategy available in systems that do not meet the [Popek and Goldberg virtualization requirements](https://en.wikipedia.org/wiki/Popek_and_Goldberg_virtualization_requirements" \o "Popek and Goldberg virtualization requirements), is where the operating system is not running user programs as native code, but instead either [emulates](https://en.wikipedia.org/wiki/Emulator) a processor or provides a host for a [p-code](https://en.wikipedia.org/wiki/P-code_machine) based system such as Java.

Internal security is especially relevant for multi-user systems; it allows each user of the system to have private files that the other users cannot tamper with or read. Internal security is also vital if auditing is to be of any use, since a program can potentially bypass the operating system, inclusive of bypassing auditing.

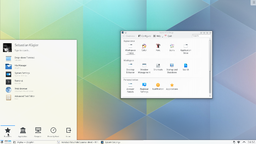
## **User interface**

[](https://en.wikipedia.org/wiki/File:Command_line.png)

A screenshot of the [Bash](https://en.wikipedia.org/wiki/Bash_(Unix_shell)) command line. Each command is typed out after the 'prompt', and then its output appears below, working its way down the screen. The current command prompt is at the bottom.

Every computer that is to be operated by an individual requires a [user interface](https://en.wikipedia.org/wiki/User_interface). The user interface is usually referred to as a [shell](https://en.wikipedia.org/wiki/Shell_(computing)) and is essential if human interaction is to be supported. The user interface views the [directory structure](https://en.wikipedia.org/wiki/Directory_structure) and requests services from the operating system that will acquire data from [input hardware devices](https://en.wikipedia.org/wiki/Input_device), such as a [keyboard](https://en.wikipedia.org/wiki/Keyboard_(computing)), [mouse](https://en.wikipedia.org/wiki/Mouse_(computing)) or [credit card reader](https://en.wikipedia.org/wiki/Credit_card), and requests operating system services to display [prompts](https://en.wikipedia.org/wiki/Command-line_interface#Command_prompt), [status messages](https://en.wikipedia.org/wiki/Status_message) and such on [output hardware devices](https://en.wikipedia.org/wiki/Output_device), such as a [video monitor](https://en.wikipedia.org/wiki/Computer_monitor) or [printer](https://en.wikipedia.org/wiki/Printer_(computing)). The two most common forms of a user interface have historically been the [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface), where computer commands are typed out line-by-line, and the [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface), where a visual environment (most commonly a [WIMP](https://en.wikipedia.org/wiki/WIMP_(computing))) is present.

## Graphical user interfaces

[](https://en.wikipedia.org/wiki/File:KDE_plasma_5.png)

A screenshot of the [KDE Plasma 5](https://en.wikipedia.org/wiki/KDE_Plasma_5)graphical user interface. Programs take the form of images on the screen, and the files, folders (directories), and applications take the form of icons and symbols. A mouse is used to navigate the computer.

Most of the modern computer systems support [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI), and often include them. In some computer systems, such as the original implementation of the [classic Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS), the GUI is integrated into the [kernel](https://en.wikipedia.org/wiki/Kernel_(computer_science)).

While technically a graphical user interface is not an operating system service, incorporating support for one into the operating system kernel can allow the GUI to be more responsive by reducing the number of [context switches](https://en.wikipedia.org/wiki/Context_switch) required for the GUI to perform its output functions. Other operating systems are [modular](https://en.wikipedia.org/wiki/Modular_programming), separating the graphics subsystem from the kernel and the Operating System. In the 1980s UNIX, VMS and many others had operating systems that were built this way. Linux and macOS are also built this way. Modern releases of Microsoft Windows such as [Windows Vista](https://en.wikipedia.org/wiki/Windows_Vista) implement a graphics subsystem that is mostly in user-space; however the graphics drawing routines of versions between [Windows NT 4.0](https://en.wikipedia.org/wiki/Windows_NT_4.0) and [Windows Server 2003](https://en.wikipedia.org/wiki/Windows_Server_2003) exist mostly in kernel space. [Windows 9x](https://en.wikipedia.org/wiki/Windows_9x) had very little distinction between the interface and the kernel.

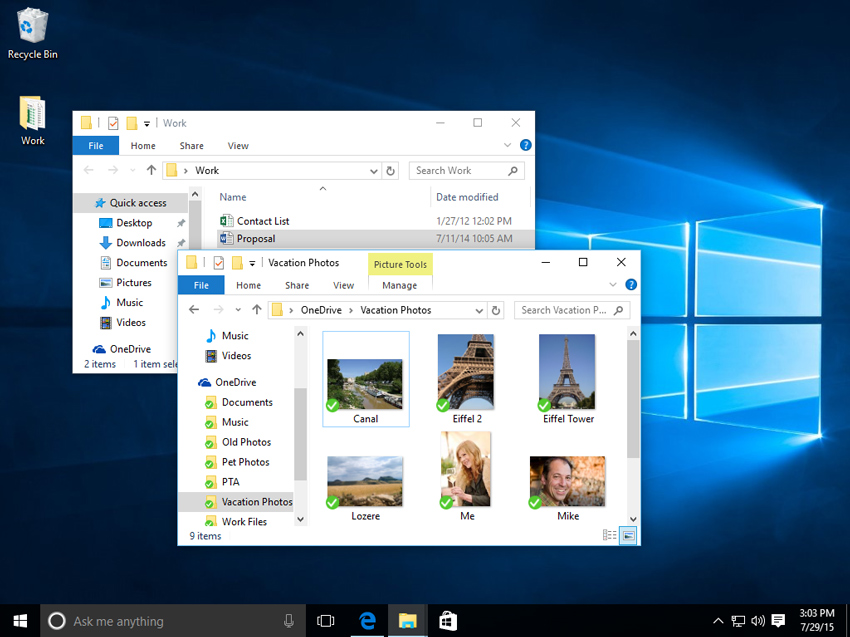
Many computer operating systems allow the user to install or create any user interface they desire. The [X Window System](https://en.wikipedia.org/wiki/X_Window_System) in conjunction with [GNOME](https://en.wikipedia.org/wiki/GNOME) or [KDE Plasma 5](https://en.wikipedia.org/wiki/KDE_Plasma_5) is a commonly found setup on most Unix and [Unix-like](https://en.wikipedia.org/wiki/Unix-like) (BSD, Linux, Solaris) systems. A number of [Windows shell replacements](https://en.wikipedia.org/wiki/Windows_shell_replacement) have been released for Microsoft Windows, which offer alternatives to the included [Windows shell](https://en.wikipedia.org/wiki/Windows_shell), but the shell itself cannot be separated from Windows.

Numerous Unix-based GUIs have existed over time, most derived from X11. Competition among the various vendors of Unix (HP, IBM, Sun) led to much fragmentation, though an effort to standardize in the 1990s to [COSE](https://en.wikipedia.org/wiki/Common_Open_Software_Environment) and [CDE](https://en.wikipedia.org/wiki/Common_Desktop_Environment) failed for various reasons, and were eventually eclipsed by the widespread adoption of GNOME and [K Desktop Environment](https://en.wikipedia.org/wiki/KDE). Prior to [free software](https://en.wikipedia.org/wiki/Free_software)-based toolkits and desktop environments, Motif was the prevalent toolkit/desktop combination (and was the basis upon which CDE was developed).

Graphical user interfaces evolve over time. For example, Windows has modified its user interface almost every time a new major version of Windows is released, and the Mac OS GUI changed dramatically with the introduction of Mac OS X in 1999

Operating systems usually come **pre-loaded** on any computer you buy. Most people use the operating system that comes with their computer, but it's possible to upgrade or even change operating systems. The three most common operating systems for personal computers are **Microsoft Windows**,**Mac OS X**, and **Linux**.

Modern operating systems use a **graphical user interface**, or **GUI** (pronounced **gooey**). A GUI lets you use your mouse to click **icons**, **buttons**, and **menus**, and everything is clearly displayed on the screen using a combination of **graphics** and **text**.



# **References**

[1] <https://en.wikipedia.org/wiki/Operating_system#Types_of_operating_systems>

[2] <https://www.gcflearnfree.org/computerbasics/understanding-operating-systems/1/>