# System software

This article is written like a personal reflection, personal essay, or argumentative essay that states a V Learn more

**System software** is software designed to provide a platform for other software.[1] Examples of system software include operating systems like macOS, GNU/Linux , Android and Microsoft Windows, computational science software, game engines, industrial automation, and software as a service applications.[2]

In contrast to system software, software that allows users to do user-oriented tasks such as create text documents, play games, listen to music, or browse the web are collectively referred to as application software.[3]

In the early days of computing[*when?*] most application software was custom-written by computer users to fit their specific hardware and requirements. In contrast, system software was usually supplied by the manufacturer of the computer hardware and was intended to be used by most or all users of that system.

The line where the distinction should be drawn is not always clear.[*according to whom?*] Many operating systems bundle[*jargon*] application software. Such

The line where the distinction should be drawn is not always clear.[according to whom?] Many operating systems bundle[jargon] application software. Such software is not considered *system software* when it can be uninstalled usually without affecting the functioning of other software. Exceptions could be e.g. web browsers such as Internet Explorer where Microsoft argued in court that it was system software that could not be uninstalled. Later examples are Chrome OS and Firefox OS where the browser functions as the only user interface *and* the only way to run programs (and other web browsers can not be installed in their place), then they can well be argued to *be* (part of) the operating system and hence system software.

Another borderline example is cloud-based software. This software provides services to a software client (usually a web browser or a JavaScript application running in the web browser), not to the user directly, and is therefore systems software. It is also developed using system programming methodologies and systems programming languages. Yet from the perspective of functionality there is little difference between a word processing application and word processing web application.

The *operating system* (prominent examples being Microsoft Windows, macOS, Linux, and z/OS), allows the parts of a computer to work together by performing tasks like transferring data between memory and disks or rendering output onto a display device. It provides a platform (hardware abstraction layer) to run high-level system software and application software.

A *kernel* is the core part of the operating system that defines an API for applications programs (including some system software) and an interface to device drivers.

*Device drivers*, including also computer BIOS and device firmware, provide basic functionality to operate and control the hardware connected to or built into the computer.

A *user interface* "allows users to interact with a computer."[4] Either a command-line interface (CLI) or since the 1980s a graphical user interface (GUI). Since this is the part of the operating system the user directly interacts with, it may be considered an application and therefore not system software.

# Difference between System Software and Application Software

Prerequisite – Software Concepts

**System Software:**

System Software is the type of software which is the interface between application software and system. Low level languages are used to write the system software. System Software maintain the system resources and give the path for application software to run. An important thing is that without system software, system can not run. It is a general purpose software.

**Application Software:**

Application Software is he type of software which runs as per user request. It runs on the platform which is provide by system software. High level languages are used to write the application software. Its a specific purpose software.

The main difference between System Software and Application Software is that without system software, system can not run on the other hand without application software, system always runs.

| S.NO | System Software | Application Software |
|---|---|---|
| 1. | System Software maintain the system resources and give the path for application software to run. | Application software is built for specific tasks. |
| 2. | Low level languages are used to write the system software. | While high level languages are used to write the application software. |
| 3. | Its a general purpose software. | While its a specific purpose software. |
| 4. | Without system software, system can't run. | While without application software system always runs. |

| | | |
|---|---|---|
| 5. | System software runs when system is turned on and stop when system is turned off. | While application software runs as per the user's request. |
| 6. | Example of system software are operating system, etc. | Example of application software are Photoshop, VLC player etc. |
| 7. | System Software programming is complex than application software. | Application software programming is simpler as comparison to system software. |