Driver

A driver is a computer software with which other software (the operating system) accesses the hardware of a device. Usually drivers are supplied with operating systems for key hardware components,
without which the system will not work. However, some devices (such as a video card or printer) may require special drivers, usually provided by the device manufacturer.

In general, a driver is not required to interact with hardware devices, it can only imitate them (for example, a printer driver that writes output from programs to a file), provide software services that are not related to device management (for example, / dev / zero in Unix, which only gives out zero bytes), or do nothing (for example, / dev / null on Unix and NUL on DOS / Windows).

Driver building approach

The operating system controls some “virtual device” that understands the standard set of commands. The driver translates these commands into commands that the device itself understands. This ideology is called
"Abstraction from hardware." For the first time in the domestic computing technology, such an approach appeared in the EC series of computers, and this kind of control software was called channel software.

A driver consists of several functions that handle certain operating system events. Usually these are 7 major events:

• Driver download. Here the driver is registered in the system, performs the initial initialization, etc.
• Unloading. The driver frees the captured resources - memory, files, devices, etc.
• Opening the driver. The beginning of the main work. Usually the driver is opened by the program as a file, by the functions fopen () on UNIX-like systems or CreateFile () on Win32.
• Read / Write: the program reads or writes data from / to the device served by the driver.
• Closing: reverse opening operation, freeing resources involved in opening and destroying the file descriptor.
• I / O control (English IO Control, IOCTL). Often the driver supports an input-output interface specific to the device. Using this interface, the program can send a special command that this device supports. For example, for SCSI devices, you can send a GET_INQUIRY command to get a device description. In Win32 systems, control is exercised through the DeviceIoControl () API function, in UNIX-like systems, using ioctl ().

Driver integration

With the development of systems combining not only the central elements of a computer, but also most of the computer’s devices as a whole, on a single board, the question has arisen of the convenience of supporting such systems, called the “hardware platform”, or simply “the platform”.

First, platform manufacturers supplied a set of separate drivers for operating systems, assembled on one carrier (usually a compact disc), then installation packages appeared, called 4-in-1 and One touch, which made it possible to simplify the installation of drivers into the system. In this case, as a rule, you can choose either fully automatic installation of all drivers, or manually select

the necessary ones. However, there was no single, well-established term for a long time.

The modern term is the Board Support Package (or “platform support package”) describing such sets of device drivers. In addition to the drivers themselves, it can, like other installation packages, contain operating system modules and programs.

Virtual drivers

Virtual device drivers are a special version of the drivers. They are used to emulate a hardware device, especially in virtualization environments, for example, when the DOS program runs on a computer running Microsoft Windows or when the guest operating system runs on Xen hosting.