Learn about emulators

The emulator is a software that simulates device or operating behavior without the actual hardware required.

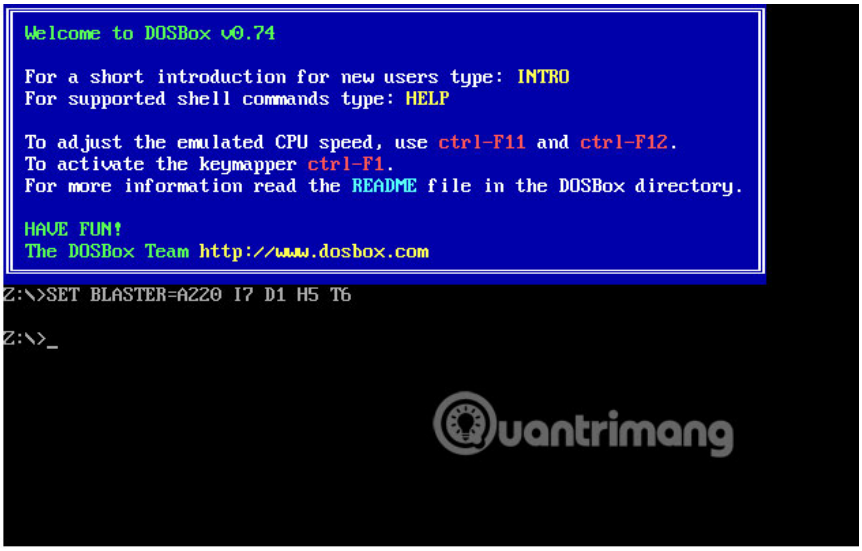
The emulator is a hardware or software that allows a computer system (called a server) to function like another computer system (called a client). The emulator usually allows the server system to run the software or use peripheral devices designed specifically for the client system. The concept of simulations refers to the ability of a computer program in an electronic device to simulate (or imitate) a program or other device.

What do you know about emulators?

1. What is emulation?

Concept of emulation

The emulator is a software that simulates device or operating behavior without the actual hardware required.



For example, the Android phone emulator will be an application that runs on a PC but looks like and acts like an Android phone. The emulator can then download and install Android applications, giving PC users access to the software that requires a real phone to do so.

The emulator is often used for two purposes. First, they are used by developers to test the system without a heap of physical equipment. Secondly, the emulator is often used to run software on computers that normally cannot be performed.

For example, Bluestacks software allows emulating an Android tablet on a Windows PC. After that, you can download and play Plants vs. Plants. Zombie 2 on that PC. Without this simulator, you can only play Plants vs Zombies 2 if you really have an Android tablet.

Benefits of emulators

Basilisk II emulates the Macintosh 68k using two **interpretation code** features (code interpreted in computer language that is understandable and executable) and **dynamic recompilation** (the system can recompile part of a chapter submit during execution).

1. Graphics quality is better than original hardware.
2. You can add additional features that the original hardware doesn't have.
3. The simulator maintains the original, sensory and behavioral interfaces of digital objects. This is as important as digital data itself.
4. Although the initial money spent developing a simulator is quite high, in the long run, this is still a cost-effective solution.
5. Reduce labor time. Instead of having to perform continuous data transfer tasks for all digital objects, when the library of past and present operating systems and application software is set up in the emulator, these This same technology will be used for all documents using those platforms.
6. Many emulators have been developed and released under the GNU General Public License through an open source environment, allowing all users to collaborate on a large scale.
7. The emulator allows software for a single system to be used on another system. For example, the PlayStation 2 exclusive video game can be played on a PC using an emulator. The emulator is particularly useful when the original system is difficult or incompatible with modern equipment (for example, old game consoles connected via analog output cannot connect to modern TVs with only technical inputs). number).

2. The simulator level

The simulator has 2 different levels: Low level and high level. The main difference of these 2 levels is how the emulation process is performed.

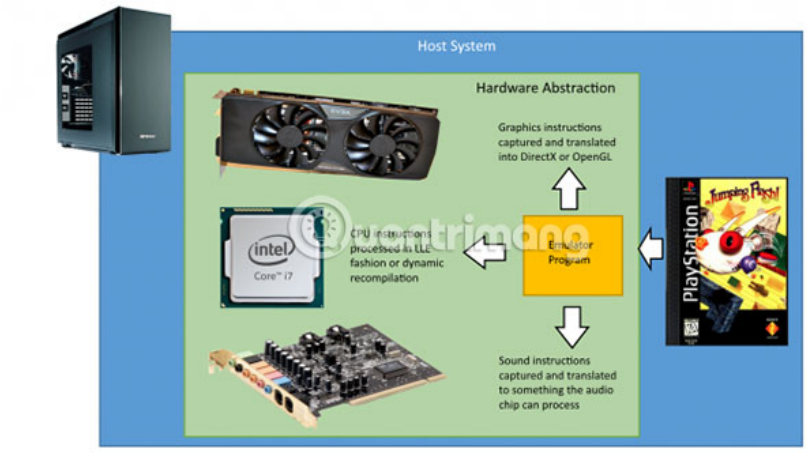
Low level emulator

Low-level emulation (LLE) mimics the behavior of simulated hardware. The server will create an environment for the application to run at the processed place, as close as possible, because the emulated hardware will perform that task. For the most accurate emulation, not only all components, but also their signals must be simulated. The more complex the system (having multiple chips or using complex chips), the lower level the implementation of the simulator will become more difficult.

Low level emulation can be achieved through hardware or software. Real hardware or something that replaces it, is within the system itself. PlayStation 3 in the first two models emulated hardware by including the actual hardware used in the PlayStation 2. Older Mac computers have an add-on card, called an MS-DOS compatible card, available. 486 processor-based systems to run x86 applications.

Simulate low-level software exactly as its name implies, simulating hardware using software. Many retro video game consoles and 8-bit home computers are emulated this way using well-known components (it's easy to find a system using MOS 6502 or Zilog Z80). An element that can make or break simulations is the frequency of synchronizing each simulated component. For example, the SNES Higan emulator works very accurately by increasing the number of times the components synchronize with each other. However, Higan requires a much faster processor than it is trying to simulate.

High level emulation



High-level emulation (HLE) takes a different approach to simulate a system. Instead of trying to emulate hardware, it simulates the function of that hardware. In the mid-90s, the concept of hardware abstraction expanded into more computer systems, including video game consoles. This allows programming to be easier because now developers do not have to re-create what is already available.

Hardware abstraction is a way to hide the complex details of hardware control. Instead, providing a set of actions that developers often use and automatically implementing all the small details involved. Typical examples are how the storage drive interface appears. Initially, if the developer wants to read the data from the drive, they must order the drive to spin, set the read / write head, select the location to read the data, and then retrieve and transfer the data to the desired location. With hardware abstraction, the developer orders 'I want to read at this location' and the firmware on the hard drive will take care of the rest. The high-level emulator takes advantage of the hardware abstraction process by finding the command (s) intended to execute in the simulator environment and leaving the server hardware to do the rest.

High level emulation has 3 main methods, simulating hardware functions.

1. Interpretation: The emulator executes each application line of code, by mimicking what each command has to do.
2. Dynamic Recompiling: The emulator looks at the instructions in the instructions from the application's processor and whether it is possible to optimize those instructions to run better on the host's processor. In stark contrast to running each tutorial one by one.

3. List blocking list: Simultaneous processors, such as GPUs and audio chips, are capable of abstracting hardware, requiring the main processor to send the command list. Here is a series of instructions for the co-processor to know what to do. The emulator can block the list of commands and turn them into something that the server can handle on the same processor. For example, the emulator's list of commands going to the GPU can be blocked and turned into DirectX or OpenGL commands to handle the host's video card.

Another example of a high-level emulator is the Java Virtual Machine (JVM). Java code is not actually compiled and run naturally on the server, but instead, the server runs a Java machine emulator in theory. Applications created for Microsoft **.NET framework** also run this way.

The performance that a high-level simulator can provide is a simulation of the Nintendo 64 on Pentium II processors in 1999, three years after the release of the console. In fact, this is the most feasible way to emulate the Xbox One to Xbox 360, although running hardware is not really remarkable.

Should a high level or a low level emulator be used?

Low level emulation is one of the most accurate ways to simulate the system in question, because it copies the behavior of the hardware. However, hardware-based emulation is not always feasible because it adds to the cost of the system. Besides, software-based emulation requires deep knowledge of the system or parts in it. This may not be possible if the reference is too 'scarce'. Software-based emulation also requires a much stronger system than the original to run applications at the same speed. Low-level emulators are often limited to older systems, prototype emulators are handling everything or smaller components of the system like the I / O driver.

On the other hand, high-level emulation allows a system with complex hardware simulated on something just a little stronger. High-level emulation can also allow a person who doesn't have deep knowledge of hardware to simulate it. However, because high-level emulation can only provide hardware functions, it cannot simulate special hardware-specific features or any tricks that developers have used. used to do something beyond the standard. Recently, however, developers have relied very little on tricks. Most of them use industry-standard APIs, making systems emulation not only easier but also capable of running applications with better performance. This is an approach to most emulators that modern systems are implementing.

3. Fields using the most emulators and shortcomings

In maintaining digital information

Emulation is a strategy in maintaining digital information to combat obsolescence. Emulator focuses on recreating the original computer environment. This process can be time-consuming and difficult, but very valuable because it provides the ability to maintain closer connection with the authenticity of digital objects.

Emulator handles the digital object's hardware and software environment and recreates it on the current computer. The emulator allows users to have access to any type of application or operating system on the current platform, enabling the software to run as in its original environment.

Jeffery Rothenberg, who initiated the use of the emulator as a strategy for maintaining digital information, said: "This ideal method will provide a long-term, scalable, only solution. need to be designed once, suitable for all use cases and capable of applying uniformly to all types of documents and media ".

He also added that the emulation could apply not only to obsolete systems, but also to systems that are unknown in the future. Practically speaking, when a new version of a certain application is released, instead of resolving the compatibility and migration issues for all digital objects created in the previous version of the application, New version, we can create an emulator for the application and allow access to all previous digital information.

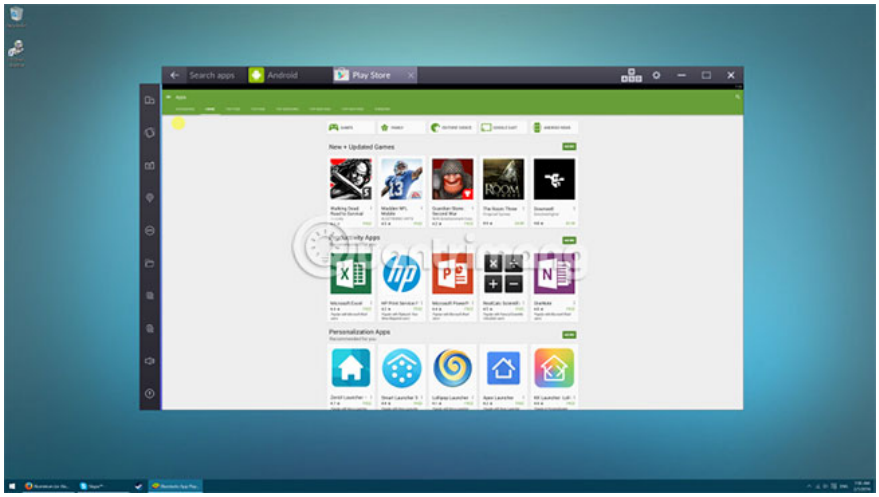
In the new media art (New Media Art)

Due to the basic use of digital formats, new media art (New Media Art) depends a lot on emulation. Artists, such as Cory Arcangel, who are dedicated to restoring backward technologies in artworks, recognize the importance of a decentralized and institutionalized process to preserve digital culture. . In many cases, the goal of simulating in new media art is to preserve a digital medium, so that it can be stored indefinitely and copied without any problems, It means that there is no reliance on hardware that is gradually degraded and obsolete. But paradox is that emulators and emulators must be designed to work on computers in the future.

The system design in the future

Emulation techniques are often used in the design and development of new systems. Emulation makes the development process easier by providing the ability to detect, reproduce, and repair design errors, even before the actual system is built. This technique is particularly useful in designing multi-core systems, where errors occur simultaneously and it is difficult to detect and repair if there is no controlled environment provided by the virtual hardware. The simulator also allows for software development before the hardware is ready to meet, helping to confirm whether the design decisions are correct.

Shortcomings need to be resolved



1. Intellectual property - Many technology providers have implemented non-standard features in the process of developing the program to build their position in the market and apply continuous upgrades to maintain competition. Although this may motivate the technology industry to grow and increase the market share of suppliers, it makes users face "nightmares" in maintaining digital information, with little talent. Data support due to the exclusive nature of hardware and software.
2. Copyright law has not yet taken effect to address the storage of documents and specifications of proprietary software and hardware in an emulator module.
3. The emulator is often used as a copyright infringement tool, because they allow users to play video games without having to buy consoles and without any effort to prevent the use of non-real copies. legal. This

leads to legal uncertainties related to emulation. Many software are programmed to refuse operation if the server knows it is an emulator. Some special video games will continue to run, but do not allow players to continue the final part of the game. These protections make the emulator design more difficult. Developers must ensure that the emulator is designed to be accurate enough to not trigger the above safeguards.

4. The emulator requires better hardware than the original system.