System image

From Wikipedia, the free encyclopedia

In computing, a **system image** is a copy of the entire state of a computer system stored in some non-volatile form such as a file. A system is said to be capable of using system images if it can be shut down and later restored to exactly the same state. In such cases, system images can be used for backup.

Hibernation is an example that uses an image of the entire machine's RAM.

Disk images [edit]

Main article: Disk image

If a system has all its state written to a disk, then a system image can be produced by simply copying that disk to a file elsewhere, often with disk cloning applications. On many systems a complete system image cannot be created by a disk cloning program running within that system because information can be held outside of disks and volatile memory, for example in non-volatile memory like boot ROMs.

Process images [edit]

A process image is a copy of its state at a given point in time. It is often used for persistence. A common example is a database management system (DBMS). Most DBMS can store the state of its database or databases to a file before being closed down (see database dump). The DBMS can then be restarted later with the information in the database intact and proceed as though the software had never stopped. Another example would be the hibernate feature of many operating systems. Here, the state of all RAM memory is stored to disk, the computer is brought into an energy saving mode, then later restored to normal operation.

Some emulators provide a facility to save an image of the system being emulated. This is often called a savestate.

Programming language support [edit]

Some programming languages provide a command to take a system image of a program. This is normally a standard feature in Smalltalk (inspired by FLEX) and Lisp, among other languages. Development in these languages is often quite different from many other programming languages. For example in Lisp the programmer may load packages or other code into a running Lisp implementation using the read-eval-print loop, which usually compiles the programs. Data is loaded into the running Lisp system. The programmer may then dump a system image, containing that pre-compiled and possibly customized code - and also all loaded application data. Often this image is an executable, and can be run on other machines. This system image can be the form in which executable programs are distributed — this method has often been used by programs (such as TeX and Emacs) largely implemented in Lisp, Smalltalk, or idiosyncratic languages to avoid spending time repeating the same initialization work every time they start up.

Similar, Lisp Machines were booted from Lisp images, called Worlds. The World contains the complete operating system, its applications and its data in a single file. It was also possible to save incremental Worlds, that contain only the changes from some base World. Before saving the World, the Lisp Machine operating system could optimize the contents of memory (better memory layout, compacting data structures, sorting data, ...).

Wikipedia The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Français
Қазақша
Lietuvių
日本語
Edit links

Although its purpose is different, a "system image" is often similar in structure to a core dump.

See also [edit]

- Disk image
- ISO image

External links [edit]

- CryoPID — A Process Freezer for Linux

Categories: Lisp (programming language) | Smalltalk programming language family | Operating system technology