Microkernel

From OSDev Wiki

Contents

Kernel Designs

Models

Monolithic Kernel
Microkernel
Hybrid Kernel
Exokernel
Nano/Picokernel
Cache Kernel
Virtualizing Kernel
Megalithic Kernel

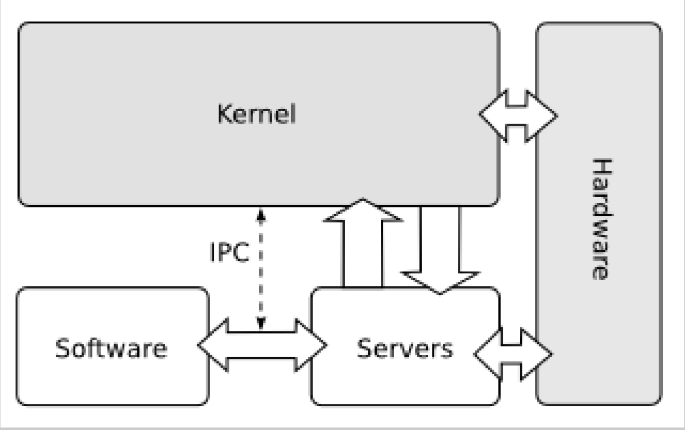
Other Concepts

Modular Kernel
Higher Half Kernel
64-bit Kernel
Bare Bones

Design

A Microkernel tries to run most services - like networking, filesystem, etc. - as daemons / servers in user space. All that's left to do for the kernel are basic services, like memory allocation (however, the actual memory **manager** is implemented in userspace), scheduling, and messaging (Inter Process Communication).

In theory, this concept makes the kernel more responsive (since much functionality resides in preemptible user-space threads and processes, removing the need for context-switching into the kernel proper), and improves the stability of the kernel by reducing the amount of code running in kernel space. There are also additional benefits for OS' that support multi-CPU computers (much simpler re-entrancy protection and better suitability for asynchronious functionality) and distributed OS' (code can use services without knowing if the service provider is running on the same computer or not). A drawback is the amount of messaging and Context Switching involved, which makes microkernels conceptually slower than monolithic kernels. (This isn't to say that a smartly designed microkernel could not beat a foolishly designed monolithic one.)



Basic overview of a microkernel

In practice things can be quite different. For example, if the filesystem crashed, the kernel would continue running, but the user would still lose some data - unless provisions were made to restart the filesystem server / daemon, and a data recovery system exists. Microkernels can be more stable, but it requires additional design work; it does not come free with the architecture. Likewise, the additional design work that has to be done to get a microkernel design right could also be spent on making a monolithic kernel preemptable.

AmigaOS, for example, was a microkernel - and an unusual one: Since the original AmigaOS had no memory protection, its messaging was as quick as it could get (passing a pointer to memory), making the AmigaOS kernel one of the fastest ever devised. On the other hand, that lack of memory protection also meant that the microkernel architecture gave no added stability (later versions did implement MMU support, but at the same speed cost that affects other microkernel systems).

One of the side-effects of using a micro-kernel design (or a modular Monolithic Kernel), is the changes needed for booting the OS. If the file system is handled by a user-space process loaded by the kernel, the kernel itself contains no code to handle file systems or storage device drivers to load the file system process in the first place. One method of resolving this is for the boot loader to load a "RAM disk image", containing the kernel and several support files (device drivers, etc).

It's also possible for an OS design to borrow concepts from both monolithic kernels and micro-kernels in order to use the benefits of either method where appropriate.

Examples

- Mach
- QNX
- L4
- AmigaOS
- Minix

See Also

Forum

- microkernels
- discussing microkernel vs modular macrokernel

External Links

- 'microkernel' google directory (http://directory.google.com/Top/Computers/Software/Operating_Systems/Microkernel/)

Recommended Reading

- The Increasing Irrelevance of IPC performance in Microkernel-based Operating Systems (http://www.google.com/search?q=The+Increasing+Irrelevance+of+IPC+Performance+for+Microkernel-Based+Operating+Systems) by Brian N. Bershad
- The Persistent Relevance of IPC performance in Microkernel-based Operating Systems (http://www.google.com/search?q=The+Persistent+Relevance+of+IPC+Performance+for+Microkernel-Based+Operating+Systems) by Wilson C. Hsieh, M. Frans Kaashoek, and William E. Weihl
- μ-Kernels Must And Can Be Small (http://citeseer.ist.psu.edu/408369.html) by Jochen Liedtke
- Microkernel-based OS Efforts (http://linuxfinances.info/info/microkernel.html) by Christopher Browne
- Microkernel Construction Notes (http://i30www.ira.uka.de/~neider/edu/mkc/mkc.html) by Raphael Neider

Retrieved from "http://wiki.osdev.org/index.php?title=Microkernel&oldid=15461"
Category: Kernel

- This page was last modified on 16 December 2013, at 11:51.
- This page has been accessed 79,146 times.