Microkernel vs. Monolithic OS Architectures

Mohammed Abualrob

Introduction

Computer operating systems have been around for more than half a century so there is not much to add to the definition. Whether you are in the software industry or the academia, I assume you know what an operating system means but in few words an OS is the piece of software that sits between computing hardware and user applications. In this article, I am going to summarize and contrast some of the most popular operating systems architectures from a high level perspective.

Generally speaking, an operating system consists of two parts: a privileged mode called kernel space and unprivileged mode called user space. The separation is a need rather than an option otherwise process protection cannot be achieved. Depending on which processes run in what space, we can classify operating systems into three main architectures: Monolithic kernel, Microkernel and Hybrid or modular kernel operating systems. Let us now explore these architectures and see the good and bad about each one of them.

Monolithic Kernel

You can think of a monolithic kernel OS as a single large static binary file process running entirely in a single address space. Basic OS services such as process management, memory management, interrupt handling, IO communication, file system, device drivers, networking, etc all run in kernel space. Entire services are loaded on boot up and reside in memory and work is done using system calls. Linux is an example on a monolithic kernel based OS.

Monolithic Kernel Advantages

Generally speaking a monolithic OS kernel is faster due to small source and compiled code size. Less code means also less bugs and security issues.

Monolithic Kernel disadvantages

Monolithic OS being a single big pile of code has disadvantages. For example, making changes is not easy and testing takes more time. It is hard to maintain, patch or extend. Bug fixing or adding new features requires the compilation of the whole source code which is a time and resource consuming process.

Microkernel

The idea behind microkernel OS is to reduce the kernel to only basic process communication and IO control and let other system services run in user space just like any other normal processes. These services are called servers and kept separate and run in different address spaces. Contrary to monolithic OS where services are directly invoked, communication in a microkernel is done via message passing (inter process communication IPC). Mac OS and WinNT are two examples on microkernel OS architecture.

Microkernel Advantages

Here some advantages to the microkernel OS architecture…

1. Service separation has the advantage that if one service (called a server) fails others can still work so reliability is the primary feature. For example if a device driver crashes does not cause the entire system to crash. Only that driver need to be restarted rather than having the entire system die. This means more persistence as one server can be substituted with another. It also means maintenance is easier.

2. Different services are built into special modules which can be loaded or unloaded when needed. Patches can be tested separately then swapped to take over on a production instance.

3. Message passing allows independent communication and allows extensibility

4. The fact that there is no need to reboot the kernel implies rapid test and development.

5. Easy and faster integration with 3d party modules.

Microkernel Disadvantages

Here are some disadvantages to the microkernel approach…

1. Memory foot print is large

2. Potential performance loss (more software interfaces due to message passing)

3. Message passing bugs are not easy to fix

4. Process management is complex

Hybrid kernel

The hybrid approach is derived from the best of both micro and monolithic kernel architectures. Instead of loading the whole thing into memory, core modules are loaded dynamically to memory on demand. One disadvantage is that a module may destabilize a running kernel.

The Linus Torvalds Andrew Tanenbaum Debate

Linus Torvalds the main guy behind Linux is a strong proponent of monolithic kernels while Andrew Tanenbaum the creator of Minix OS is a supporter of microkernel. Tanenbaum says that microkernel is a better design principle especially when reliability is of uttermost importance. According to Tanenbaum, reliability scores over performance gains because microkernels allow building self healing systems.

Conclusion

I will conclude this article by saying there is no OS architecture that is better than the other in the general sense. I think comparing monolithic operating system is like comparing two different types of transportation for example a car and a bus where each meets different needs.

Search Terms...

- monolithic vs microkernel
- difference between monolithic kernel and microkernel
- monolithic kernel vs microkernel
- microkernel vs monolithic kernel
- difference between microkernel and monolithic kernel
- microkernel vs monolithic
- advantages of monolithic kernel
- advantages and disadvantages of monolithic kernel

- difference between monolithic and microkernel
- disadvantages of microkernel

Related Posts ...

- Difference Between User Level Threads and Kernel Level Threads
- File Allocation Methods
- Difference between Multiprogramming, Multitasking, Multithreading and Multiprocessing
- Difference Between System Call, Procedure Call and Function Call
- Disk Scheduling Algorithms