多内核：可扩展多核系统的新操作系统架构

随着不断变化的技术推动摩尔定律限制，处理器体系结构正在变得越来越多样化，以切换异构性，并朝着可扩展架构迈进，以适应高性能应用。传统的单片操作系统对解决这种可扩展性问题和优化各种硬件结构提出了巨大的挑战。本文中的作者尝试通过在内核之间使用显式消息传递以及跨内核复制内核状态来解决此问题，而不是共享内存模型。他们的另一个主要目标是使此操作系统硬件保持中立，因此不适用于任何机器架构。

于是本文作者提出了 Multikernel 及其原型 Barrelfish，这是一种新的操作系统体系结构，它继承了分布式系统的主要特性和见解，旨在实现跨不同多核硬件系统的更好的可扩展性。除了设计原则外，本文还介绍了 Barrelfish 的实现细节及其对各种工作负载的评估、证明。

Multikernel 模型遵循三个设计原则：第一、明确所有核心间通信，而不是在通信内核之间共享内存，所有通信都使用消息传递完成。这提供了模块化，因为通信已经完成了定义良好的接口。第二、使 OS 结构硬件中立，如上所述，这使得 OS 具有可扩展性和未来的可行性。消息传递机制和硬件接口是依赖于硬件的；第三、将状态视为复制而不是共享，每个内核都有自己的状态副本。这些状态之间保持一致。这个副本靠近核心可用，因此具有更好的性能。

他们的实验表明，通过扩展线程/内核数量，消息传递证明性能比共享内存模型更好。系统结构具有提供硬件结构访问权限的低级别 CPU 驱动程序和负责监视内核之间的 RPC 的用户监视器。在内核之间复制内核状态会使系统中具有不同内核的可能性，从而不能针对单个体系结构优化操作系统。

arrelfish 原型已经在诸如 Intel Xeon 和 AMD Opteron 芯片等许多多处理器系统上进行过测试。虽然没有强调他们使用的方法论，但论文中提到了很多绩效结果。相同处理器核心和不同内核上的用户级别 RPC 调用性能数字表明，消息传递延迟和吞吐量仍处于可接受的范围内。针对不同数量的内核，针对不同的消息传递策略和操作系统比较了 TLB 关闭和内存取消映射性能。结果表明，Barrelfish 虽然对较低的核心数量表现不佳，但随着核心数量的增加而扩大。AMD 计算机上的计算工作负载的 Barrelfish 性能证明与 Linux 操作系统类似。