

**Question 1: Application can gain a better performance of memory and file system in exokernel compared with microkernel or monolithic kernel, why? And what's the price of performance profit?**

为什么性能更好：因为相比于 microkernel 与 monolithic kernel, exokernel 的用户态权限最大，导致应用可以直接调用物理资源，例如文件系统，内存虚拟地址空间等

代价：exokernel 一旦出错，很难去定位 bug，维修人员不知道用户运行的是哪个 OS

设计 exokernel 的接口很复杂，一致性不好

**Question 2: Exokernel grants the applications more control over hardware resource. How does exokernel protect application against each other? And how to understand this goal of paper: to separate protection from management?**

**Protect application against each other:** Exokernel 系统本身只保证被请求的资源当前是空闲的，应用程序就允许直接访问它。每个用户进程都拥有一个虚拟机，可以运行自己的操作系统。Exokernel 会防止其他用户进程访问到进程的资源。就是 Exokernel 一旦把资源给我了，在我释放这部分资源之前，其他人不可以访问到我手上的资源。

**Understand the goal:** 内核资源管理的目的应该是保护所必需的功能，不应该做更多的事情。内核只负责保护，如何管理这些资源就交给应用本身。因为每个应用要做的东西，要在什么资源上做什么操作都不一样，这样把操作资源的权限给应用，才更方便性能更好。

**Question 3: Open question - do you think why has not exokernel been as popular as monolithic kernel(Linux) and Hybrid kernel(Windows NT)?**

libOS 现在也还不错。为什么之前不火呢？感觉是因为虽然应用的权限变大了，但是相对的，开发应用的成本就变高了，毕竟还要自己手写管理资源的代码，Exokernel 的接口实现起来也比较难，程序员懒，所以 Exokernel 不火。