

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?

The high performance is due to the high freedom and flexibility provided by Exokernel. In Exokernel, it allocates the basic physical resources to application and let applications manage resources by themselves instead of invoke through a bunch of abstraction.

And the price is that kernel have to provide varieties of library operating system to application. And if necessary, app developer have to custom their own library operating system. So it adds difficulty for developing.

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?

- Secure Bindings: is a protection mechanism that decouples authorization from the actual use of a resource.
- Visible Resources Revocation: a way to reclaim them and break their secure bindings.
- Abort Protocol: Abort protocol will kill any library operating system and its associated application that fails to respond quickly to revocation requests.

They protect resources but delegate management to application.

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

1. Linux and Windows NT, they got a long history and many applications are based on them. It takes a lot of effort to shift to new platform.
2. Exokernel has less generality than monolithic kernel and hybrid kernel, which means it's harder to get popular because it needs more efforts to develop specific library systems and maintain them.