Exokernel: An Operating System for Application-level Resource Management – Critique

- Shashidhar Kumar Thangallapelli (862395355)

Summary:

Conventional Operations Systems usually have a centralized resource manager which limits the flexibility, performance and functionality of applications. These applications run on a Virtual Machine with fixed high-level abstraction. To address this particular problem and to provide application-level control of physical resources an *Exokernel* is introduced. This includes an exokernel and an untrusted Library Operating System. This Exokernel securely binds hardware resources to the library OS through a low-level interface. Now the library operating system uses this interface to implement system policies/objects. By doing such it performed ten to hundred times faster than the Ultrix operating system which is a UNIX based operating system. The paper continues to provide more details regarding the architecture of "Aegis" the exokernel and "ExOS" the Library Operating System.

Strengths:

- Of Course, the main strength of this architecture is, it has high flexibility and functionality through allowing specializations, extensions and replacing existing level of abstraction.
- Providing higher performance by resolving the conflicts between applications without the involvement of kernel architects.
- Very good in terms of portability, compatibility and easy to implement.

Weaknesses:

- Not having any abstraction in the operating system is not good, as it requires for the architect while implementing this OS to write these abstractions which makes the application significantly complex to implement.
- The paper just measures the metrics by use of prototypes but not a real-world Operating System. This in my opinion does not feel complete to what it's trying to achieve, although the direction where it's going is correct.

Other Comments and discussion:

The idea of having a minimal kernel i.e., the exokernel for having that specialized functionality and flexibility for some applications in regards to hardware resources is fascinating. This paper again tries to achieve the similar goals and ambitions of SPIN OS such as extensibility, better performance, specialized functionality, it takes rather a different approach for achieving these. The comparative metrics or results shown in the paper are not any better than other proposed architectures of similar kind. As this is just a prototype, many of the security protocols and abstractions are not implemented in the experiment, thus I feel the results or metrics are not the final ones to consider. Further having all these might impact the performance and decrease the efficiency of the final product.