1. 三篇论文的核心思想总结

现代数据中心对弹性、效率和安全的极致追求，正将传统操作系统的设计推向极限。LegoOS[1]、FlexOS[2] 和 DBOS[3] 这三项研究，共同指出了当前单体服务器和宏内核架构的根本性缺陷。LegoOS 的核心批判在于硬件的僵化耦合。在传统服务器中，CPU、内存和存储被“焊死”在一起，形成了“资源孤岛”。这种设计不仅导致了严重的资源利用率低下，例如 CPU 密集型应用耗尽了处理器却闲置了大量内存，同时也带来了弹性差、难以升级和故障域大的问题。

在 LegoOS 揭示硬件层面的束缚时，FlexOS 和 DBOS 则分别从软件的安全性和集群管理的复杂性上发起了挑战。FlexOS 指出，当前 OS 在设计时就锁定了特定的安全和隔离策略，如固定的用户/内核态分离。这种僵化的安全模型无法适应现代应用多样化的需求——有的需要极致性能，有的需要强隔离；它也难以在不牺牲性能的前提下，快速响应新硬件隔离机制的出现或新漏洞的发现。DBOS 则将视角拉高到整个数据中心，它认为当 OS 扩展到集群规模时，其调度、文件管理和监控等服务本质上已演变为一个大规模的“大数据”状态管理问题。而现有 OS，如 Linux，本身不具备集群管理能力，导致上层，如 Kubernetes，与下层 OS 成为“两层皮”，在集成度、性能和功能上，例如事务性和高可用，都存在巨大鸿沟。

2. 我对操作系统未来发展方向的理解

面对传统架构的困境，这三篇论文共同勾勒出了未来操作系统的几个核心发展原则。首先是“解耦”与“模块化”。LegoOS 为此提出了被称为 Splitkernel 的“分裂式内核”架构，将 OS 功能分解到各个解耦的硬件组件监视器上。例如，内存管理逻辑包括分配和映射，被下放到网络连接的内存组件上执行。FlexOS 则在软件层面实现了“功能与隔离策略的解耦”。它通过高度模块化的 LibOS 设计，允许用户在编译或部署时，像搭乐高一样“定制”OS 的隔离方案，例如是使用轻量级的 MPK 还是强隔离的 VM/EPT。

其次，未来 OS 必须从“单机”思维转向 Datacenter-scale 的“数据中心即计算机”思维。LegoOS 的全局管理器，如 GMM，负责在整个集群范围内协调资源，而 DBOS 则更进一步，试图从根本上解决集群范围内的状态管理难题。

DBOS 的“万物皆为表”理念，即 Everything is a table，构成了未来 OS 的第三个核心原则：将“状态管理”作为 OS 的核心。DBOS 提议将一个分布式事务 DBMS 作为 OS 的基座。如此一来，OS 的调度、文件系统、IPC 和监控等服务，全部可以被实现为对该数据库的标准查询和事务。这种设计使得 OS 原生具备了事务性、高可用性和强大的可查询性——而这些能力在当前“Linux + K8s”的栈中是极其复杂和脆弱的。

3. 结合我的研究领域（CXL 共享内存）的展望

我所研究的 CXL 共享内存领域，正是上述三大趋势的完美交汇点。CXL 技术[4]，尤其是 CXL.mem 和 CXL switch 的出现，使得 LegoOS 所畅想的“硬件资源解耦”从基于 RDMA 的模拟，延迟约 6.5 µs，变为了低延迟且缓存一致性的物理现实，其延迟仅几百 ns。CXL 内存池正是 LegoOS 的内存组件真正形态。因此，一个 CXL 感知的“分裂式内核”将成为必然：

[1]Shan et al.，“Legoos: A Disseminated, Distributed OS for Hardware Resource Disaggregation”.
[2]Lefeuvre et al.，“Flexos: Towards Flexible OS Isolation”.
[3]Skiadopoulos et al., DBOS: A Dbms-Oriented Operating System.
[4]Das Sharma et al.，“An Introduction to the Compute Express Link (Cxl) Interconnect”.

OS 的进程监视器运行在主机上，而内存监视器则可以下沉到 CXL 内存池的控制器上运行，例如 ARM 核，专门负责池内内存的分配、共享和权限管理。我所关注的 CXLfork[5] 项目中的全局内存管理器 GMM，其职责就与 LegoOS 的内存监视器设计不谋而合。

同时，CXL 带来的细粒度内存共享，例如 VM 间或进程间共享，也引入了新的安全挑战，这使得 FlexOS 的“灵活隔离”思想至关重要。一个 CXL OS 必须为不同的租户和应用提供可定制的安全策略。例如，信任的应用间可以选择高性能的直接共享模式；而对于多租户场景，OS 则必须在部署时应用 FlexOS 的理念，利用 IOMMU 或 Intel MPK 等硬件机制，在共享的 CXL 内存池中划分出相互隔离的“安全域”，在安全和性能之间实现动态权衡。

最后，当 CXL 交换机将成百上千的主机连接到 PB 级的内存池时，整个 CXL Fabric 的状态管理将变得极其复杂。此时，DBOS 的“数据库内核”思想将成为管理 CXL Fabric 的终极方案。我们需要一个“CXL Fabric 管理器”，它本质上就是一个 DBOS：将所有 CXL 内存池、交换机、主机的状态信息，例如拓扑、延迟、容量和权限，全部“表格化”。未来 OS 的内存分配器将不再是简单的本地调用，而是向这个“CXL 状态数据库”发起一次事务性查询，例如：“请原子性地为 VM-A 在延迟小于 250 ns 的内存池中分配 50GB 内存，并设置对 VM-B 的只读共享权限”。

[5] Alverti et al.，“Cxlfork: Fast Remote Fork over Cxl Fabrics”.

参考文献

Alverti, Chloe, Stratos Psomadakis, Burak Ocalan, Shashwat Jaiswal, Tianyin Xu, and Josep Torrellas. “Cxlfork: Fast Remote Fork over Cxl Fabrics.” In “Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2.” Special issue, Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, 2025, 210–26.

Das Sharma, Debendra, Robert Blankenship, and Daniel Berger. “An Introduction to the Compute Express Link (Cxl) Interconnect.” ACM Computing Surveys 56, no. 11 (2024): 1–37.

Lefeuvre, Hugo, Vlad-Andrei Bădoiu, Alexander Jung, et al. “Flexos: Towards Flexible OS Isolation.” In “Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems.” Special issue, Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2022, 467–82.

Shan, Yizhou, Yutong Huang, Yilun Chen, and Yiying Zhang. “Legoos: A Disseminated, Distributed OS for Hardware Resource Disaggregation.” In “13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18).” Special issue, 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), 2018, 69–87.

Skiadopoulos, Athinagoras, Qian Li, Peter Kraft, et al. DBOS: A Dbms-Oriented Operating System. VLDB Endowment, 2021.