

Summary

The paper presents an extensible microkernel which provides control over system resources to applications dynamically by adapting system to the application requirements.

Main Points

The paper claims that services provided by Operating System are static and have not evolved at the same rate as applications to match its resources demand and usage patterns, which prevents application from performing well. For e.g: Conventional Disk-buffering and Paging mechanism was inappropriate for the Databases Applications.

SPIN provides an application specific services which are loaded into the kernel at runtime and enable resources to be managed efficiently and safely. These services are implemented with language level abstractions provided by the kernel which ensures security and compatibility.

The SPIN architecture consists of a set of core services and extension services that run in the kernel space and a set of applications that use those services. SPIN Architecture provides infrastructure to safely combine kernel and application code, low-cost mechanisms for procedure calls and does not depend on hardware services.

Core services provides control to applications via interfaces over physical and virtual memory resources and interfaces for scheduling, concurrency and synchronization of its threads. Extensibility is provided using events and handlers. Extensions install handlers for an event in the kernel, these events can be software or hardware exceptions. Handlers are called when an event it is binded to occurs. Handlers implementation can be application dependent to meet its requirements.

SPIN architecture is created with **Modula-3** programming language as it provides type safety, interfaces, objects, threads, exceptions and automatic storage management. The system allows applications themselves to be written in any programming language which runs in user address space.

SPIN performance is compared with **DEC OSF/1** and **MACH3.0**. SPIN outperforms both kernels in terms of procedure calls, thread management, memory management, latency and bandwidth and End-to-End application performance. The reason being SPIN can easily customize its system requirements based on application using the application specific services and thus outperform other kernels.

Limitations/Critique

The Drawbacks of SPIN microkernel can be mandating a portion of application code to be written in specific programming language which can be problem for application programmers. The other issue is installation of application extensions into kernel, it increases the efficiency of the application but add redundancy if multiple application uses same policy for thread and memory management.

Proposed Extension

SPIN can be improved by providing mechanism to use application specific services by multiple applications and thus reduce the code redundancy. Use software fault isolation which can enable application code to be written in any language and installed into kernel.