Problem :

Some applications that need to access resources are unable to run with high performance due to limitations of interfaces and implementations of a general purpose Operating System. Operating systems fail to meet the demands of such applications resulting in lower performance of such applications. Some OS have been specialized for one class of applications, but this approach reduces the performance of other classes of applications.

Key Idea :

The key idea of this paper is to use a monolithic kernel which provides very fine-grained control over kernel and resources using extensions services. Extensions have access to kernel and they can be used by applications to improve performance of the applications. Modula-3 is used to write these extensions and modularity is enforced by the compiler which restrict extensions from accessing memory and privileged instructions unless given the permission explicitly. SPIN also includes framework for memory and processor resources in its core services.

Strengths :

* SPIN uses dynamically linked modules for quicker access to the resources. The use of Modula-3 for extensions provides type safety. Security restrictions and interface checking are enforced by the compiler.
* The idea of isolating extensions is also proposed. If a extension fails to utilize the interface, it only results in failure of applications using that extension.
* The OS allows applications to define specialized fault handling extensions which avoids crossing the user-kernel boundary and implement the functionality that is required.

Weakness :

* SPIN highly depends on Modula-3 language and compiler for safety which is not good. The paper says that object files that are signed by Modua-3 compiler are identified as safe, but the sign can be forged by an application developer, then the malicious extension can be loaded to the kernel to exploit the system.
* The implemention of automatic storage management is bad. The automatic garbage collector is triggered more frequently when a application which require large memory is running. This frequent triggering has its effects on the entire system.

Thoughts on the paper:

SPIN OS did solved the problem of finding right balance between generalizing and specializing the system for all the applications. I believe the extensibility approach solves the problem efficiently, providing flexibility to the applications to use the extensions. The idea is somewhat similar to kernel modules being loaded in running kernel in a linux system, although SPIN providers more privileges and flexibility in terms of what a extension can do.With use of SPIN, various types of applications which require resources extensively can be developed, as SPIN ensures high performance of such applications. It also showcased depending on compiler and programming language to provide safety in operating systems, but I doubt relying solely on a programming language for typechecking, as it has the language’s limitations.