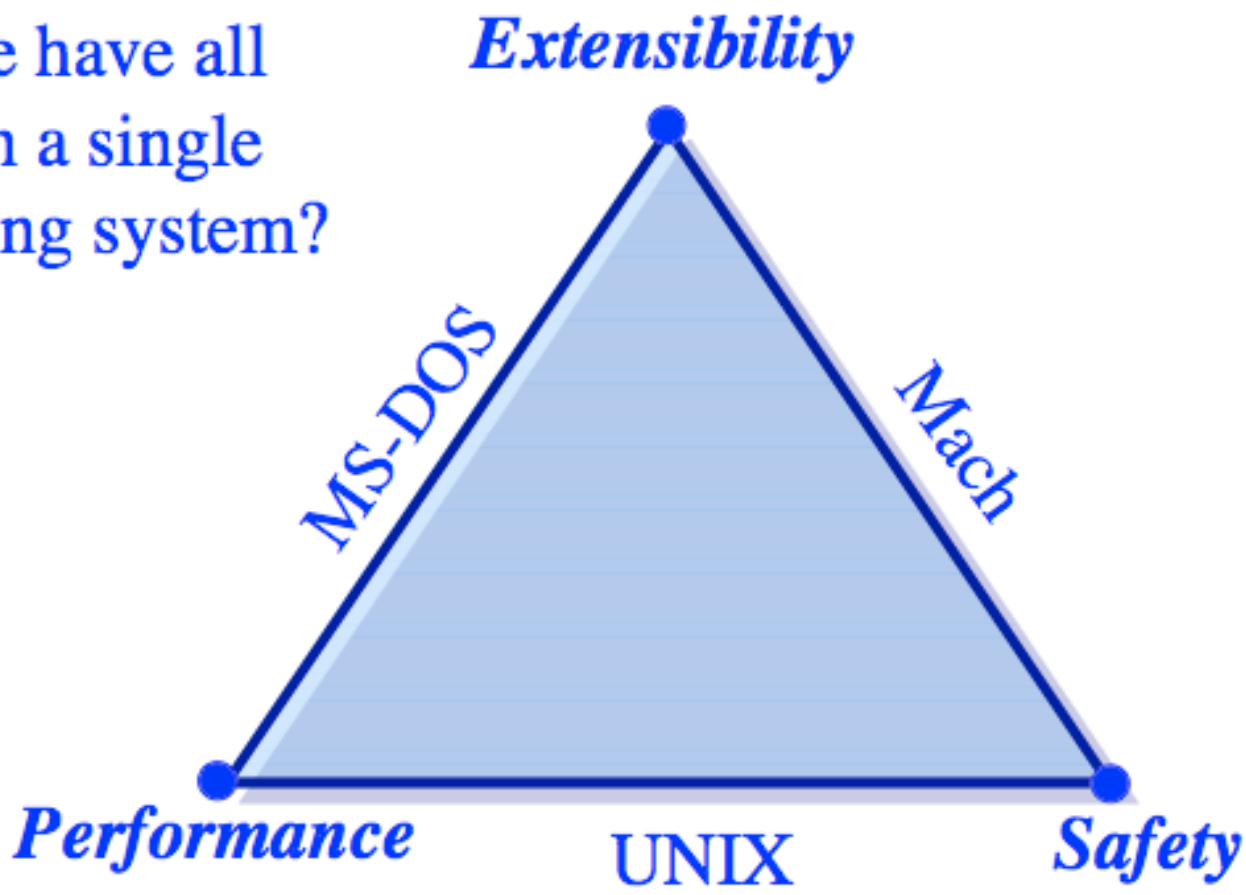
- **Extensibility safety and performance in the SPIN operating system Bershad, B. N., Savage, S., Pardyak, P., Sirer, E. G., Fiuczynski, M. E., Becker, D., Chambers, C., and Eggers, S. SOSP '95**
  - This paper was published in the same SOSP as Exokernel

Goals

- Extensibility: Applications can dynamically extend system to provide specialized services
- Safety: Kernel is protected from actions of extensions
- Performance: Extensibility and safety have low cost
- Approach:
  - Put extension code in the kernel: Cheap communication
  - Use language protection features: Static safety
  - Dynamically interpose on any service: Fine-grained extensibility

Can we have all three in a single operating system?



Comparable system

- Modules/DLLs in OS
  - Modules are per system and not per application
  - Modules offer no protection – blue screen of death if things go wrong inside a module
- Exokernel
  - Some protection boundary crossing, optimized in Aegis and might not be general purpose
- SPIN relies on Modula3 for performance and static analysis of code

- Safety: Modula3, dynamically restricts access to private interfaces at linktime
- Extensibility: Event based model, replace event handler
- SPIN implements events that cannot be safely extended
  - Notion of trust, Modula gives you safety but not trust

Discussion

- Language based approach