Extensibility: SPIN and exokernels

Robert Grimm
New York University
The Three Questions

- What is the problem?
- What is new or different?
- What are the contributions and limitations?
OS Abstraction Barrier

- Fixed high-level abstractions
  - Hurt application performance
  - Hide information
  - Limit functionality
- Examples
  - Buffer cache management
  - Persistent storage
Goals

- Extensibility
  - Applications introduce specialized services
- Safety
  - Kernel, applications, services are protected
- Performance
  - Extensibility and safety have low cost
Why Is This Hard?

Can we get all three in a single system?
Two Approaches
SPIN Approach

- Put extension code in the kernel
  - Cheap communication
- Use language protection features
  - Static safety
- Dynamically impose on any service
  - Fine-grained extensibility
The Big Picture
Modula-3

- Type-safe programming language
- Interfaces
- Garbage collection
- Other features
  - Objects, generic interfaces, threads, exceptions

- Most of kernel written in Modula-3
- Extensions must be written in Modula-3
- User-space applications written in any language
Safety

- Capabilities
  - Simply a pointer
    - Can we pass capabilities to user-land?

- Protection domains
  - Language-level
  - Limit visibility of names
  - Enforced at dynamic link time
Extensibility

- Extension model
  - Events
  - Event handlers
  - Guards

- Mechanism
  - Event dispatcher
  - Common case: procedure call
Core Services

- Memory management
  - Physical addresses
  - Virtual addresses
  - Translations

- Thread management
  - Signals to scheduler
    - Block, unblock
  - Signals to thread manager
    - Checkpoint, resume
Performance

- It works
Exokernels Approach

- Make the application do it!
Exokernels Approach (again)

- Separate protection and management
- Expose allocation
- Expose names
- Expose revocation
- Expose information
At The Core

- Processor time slices
- Processor environments
  - Hardware exceptions (Aegis, Xok)
  - Timer interrupts (Aegis, Xok)
  - Protected entries (Aegis, Xok)
- Addressing
  - Aegis: Guaranteed mappings, applications notified of TLB misses
  - Xok: Hardware page tables, applications specify mappings
- Hierarchical capabilities (Xok only)
- Book keeping

Aegis: MIPS-based DECstations
Xok: x86-based PCs
Case Study: The Disk

- Problem
  - How to store meta-data?
    - Ownership of disk blocks
- Failed approaches
  - Simple capabilities
  - Self-descriptive meta-data
  - Template-based descriptions
The Disk (cont.)

- Untrusted deterministic functions
  - Programmatic templates
- Shared data
  - Buffer cache registry
- Ordered disk writes
  - Ensure consistency after crash
Performance

- It works
- It scales
Issues

- SPIN
  - Trusted compiler
  - Resource control
Issues (cont.)

- Exokernels
  - Extension model
  - Downloaded code
    - Wakeup predicates
    - Dynamic packet filters
    - Application-specific handlers
    - Untrusted deterministic functions
  - Complexity of disk management
What Do You Think?