

# CS 5204 Operating Systems Lecture 12

Godmar Back



## Announcements

- Change in plans:
  - SPIN paper presentation on Monday
  - Can submit evaluations now or until Monday



CS 5204 Fall 2005 11/10/2005 2

## Extensible OS Research in 90s

- Idea: Extensibility, extensibility, extensibility
- Number of projects explored different approaches
  - Exokernel (MIT)
  - SPIN (UW)
  - VINO (Harvard)
  - Fluke/OSKit (Utah)
  - Scout (Arizona)
  - L4 (Liedtke/IBM)
  - Nemesis (U Cambridge)



CS 5204 Fall 2005 11/10/2005 3

## Exokernel: exterminate all OS abstractions

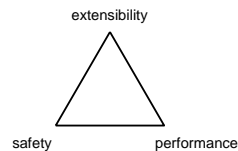
- *The defining tragedy of the operating systems community has been the definition of an operating system as software that both multiplexes and abstracts physical resources. The view that the OS should abstract the hardware is based on the assumption that it is possible both to define abstractions that are appropriate for all areas and to implement them to perform efficiently in all situations. We believe that the fallacy of this quixotic goal is self-evident, and that the OS problems of the last two decades can be traced back to it. The solution we propose is simple: complete elimination of OS abstractions by lowering the operating system interface to the hardware level.*



CS 5204 Fall 2005 11/10/2005 4

## SPIN

- *Protection is a software issue*
  - First use of a type-safe language to build a kernel (Modula-3)
  - Idea: applications can download code into the kernel *safely*



CS 5204 Fall 2005 11/10/2005 5

## SPIN (cont'd)

- How to manage extensions
  - What if they run for too long?
  - How to unload extensions?
  - How to manage their resources (memory, etc.)?
- How to foresee what extensions might want to extend?
  - Size of exposed interface
- How to arbitrate among extensions?
- Type-safe language issues
  - garbage collection
  - language domain crossings (M3 to C and vice versa)



CS 5204 Fall 2005 11/10/2005 6

## Language-based OS

- If protection is a software issue, why not do away with hardware protection altogether?
- Inferno OS system by Bell Labs
  - Limbo: typesafe language
  - DIS: virtual machine
- Early aspirations of Java
  - Java Operating Systems

## Java Operating Systems

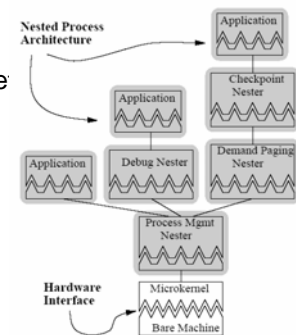
- Two views:
  - A fully-fledged operating system written in Java that runs Java applications.
    - Sun's first JavaOS, [Squawk JVM](#)
  - A JVM that provides an OS-like environment for Java applications. The so enriched JVM can run on top of any OS.
    - J-Kernel, MVM, KaffeOS

## VINO

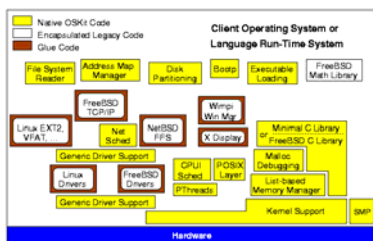
- Idea: extensions (“grafts”) are “safe overrides” for functionality at (C-) function level
- Use a transaction mechanism to recover from misbehaved extensions
  - Full transaction mechanism inside the kernel
  - Based on monitoring load/stores from C++ programs

## Fluke

- Microkernels meet recursive virtual machines
- Idea: processes could create whatever abstractions they wished for their descendants



## OSKit



- Idea: build OS you want from components

## IO-Lite/Flash-Lite

- Opposing View: Druschel et al HotOS'97
  - [Extensible Kernels are Leading OS Research Astray](#)
- Extensibility mostly busy solving problems it created
- Instead, develop, test, decide, and integrate into commodity OS
- Case in point: IO-Lite & Flash server
  - Zero copy I/O system & event-based webserver
  - Applications control I/O buffer inside kernel

## Discussion

- What did we learn from extensible OS research?
  - Is extensibility as important as claimed?
  - If not, what is?
- Is developing entire new OS still feasible today in an academic environment?
  - Is doing so necessary for exciting systems research?