Extended OS
OS is an extended virtual machine

- Multiplexes the “machine” between applications
  - Time sharing, multitasking, batching
- Provided a higher-level machine for
  - Ease of use
  - Portability
  - Efficiency
  - Security
  - Etc....
JAVA – Higher-level Virtual Machine

• write a program once, and run it anywhere
  – Architecture independent
  – Operating System independent

• Language itself was clean, robust, garbage collection

• Program compiled into bytecode
  – Interpreted or just-in-time compiled.
  – Lower than native performance
Issues

- Legacy applications
- No isolation nor resource management between applets
- Security
  - Trust JVM implementation? Trust underlying OS?
- Performance compared to native
Is the OS the “right” level of extended machine?

- Security
  - Trust the underlying OS?
- Legacy application and OSs
- Resource management of existing systems suitable for all applications?
- What about activities requiring “root” privileges
Virtual Machine Monitors

- Provide scheduling and resource management
- Extended “machine” is the actual machine interface.
IBM VM/370

Diagram:
- Virtual 370s
- System calls here
- Trap here
- I/O instructions here
- Trap here

Layers:
1. I/O instructions here
2. CMS
3. CMS
4. CMS
5. VM/370
6. 370 Bare hardware
Advantages

- Legacy OSes (and applications)
- Concurrent OSes
  - Linux – Windows
  - Primary – Backup
- Security
  - VMM (hopefully) small and correct
- Performance near bare hardware
  - For some applications
Virtual R3000???

• Interpret
  – System/161
    • slow
    – JIT dynamic compilation

• Run on the real hardware??
Issues

- Privileged registers (CP0)
- Privileged instructions
- Address Spaces
- Exceptions (including syscalls, interrupts)
- Devices