Virtual Machines

Witawas Srisa-an

CSCE496/896: Embedded Systems Design and Implementation

The Basics

- Process view
  - bare machine with memories, registers, system calls for I/O
  - OS + hardware
- OS view
  - hardware alone

Basics

- Virtual machines execute software in the same manner as the machine for which the software was developed.
  - real-machine
  - virtualization software
  - different resources than the real machines

Basics

- Virtualization has two parts
  - mapping of virtual resources or states to real resources
  - the use of real machine instructions and/or system calls

Basics

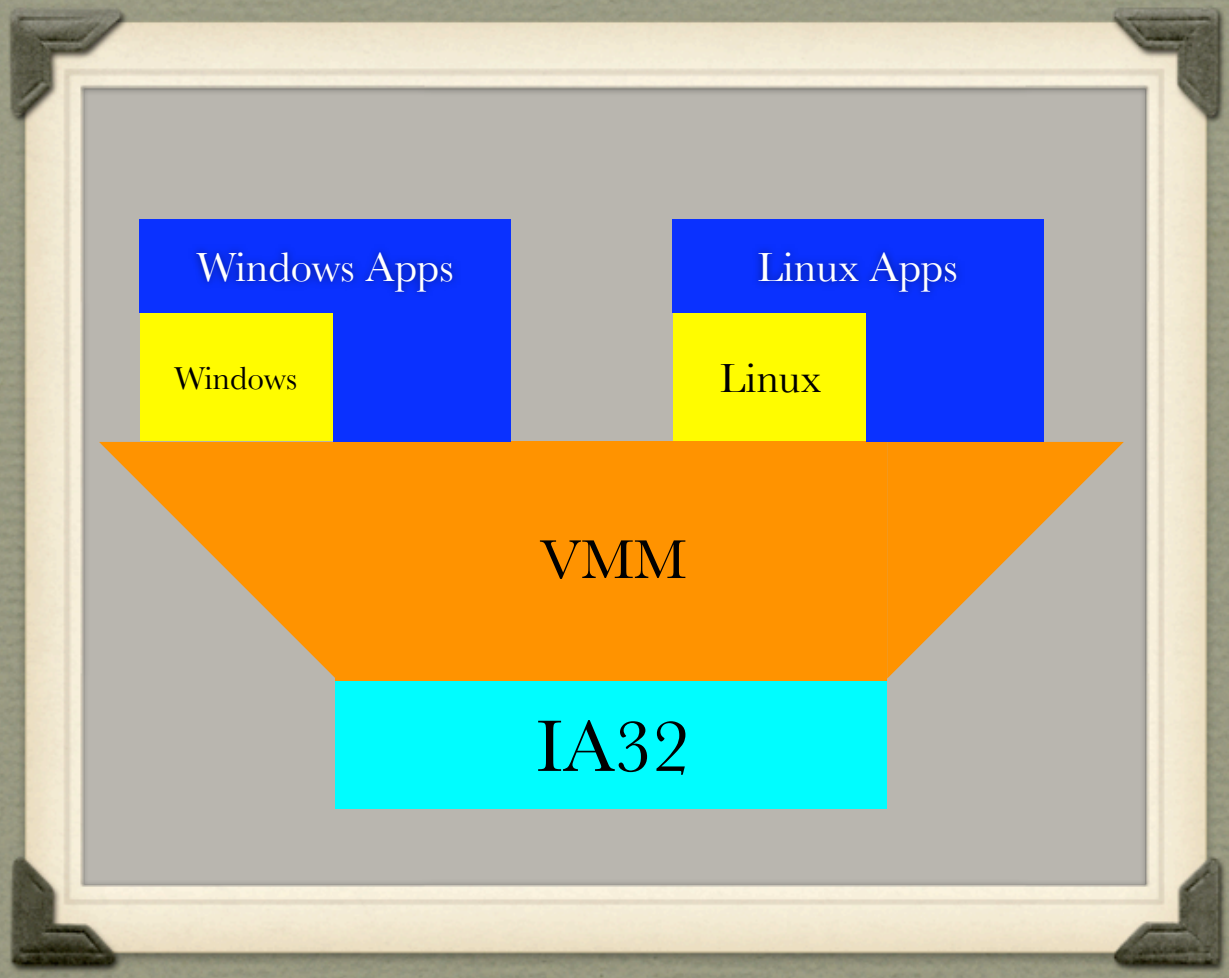
- Process level VMs
  - support an individual process
- System level VMs
  - provide complete system environments (e.g. OSs and multiple processes)

Basics

- Process VMs
  - host, runtime, guest
    - e.g. Rosetta

Basics

- System VMs
  - host, VMM, guest
    - e.g. Parallel, virtual PC, Xen



Basics

Process VMs

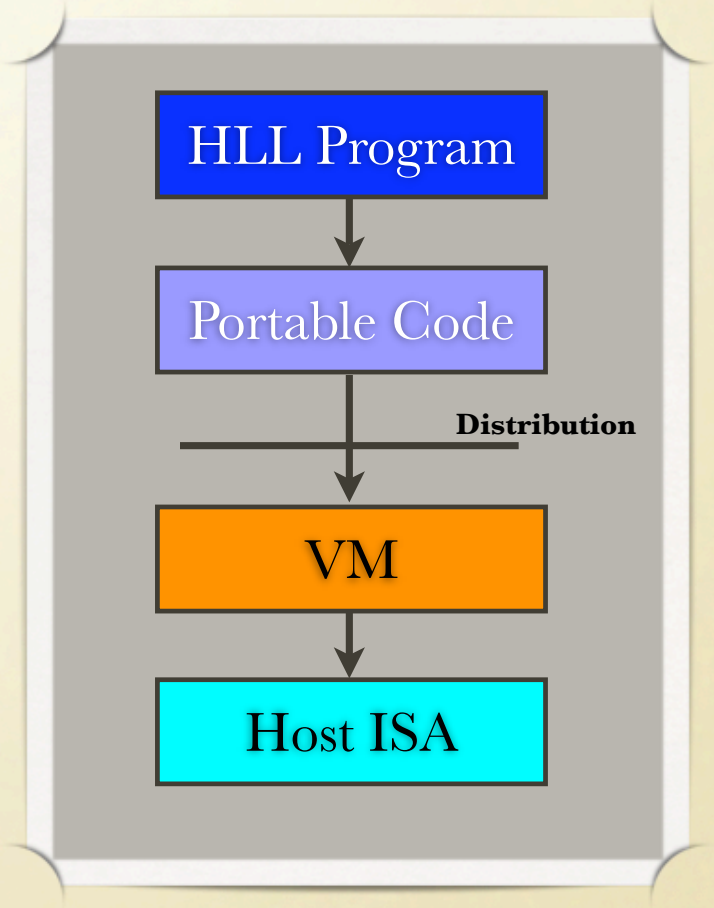
- Multiprogramming
- Emulators and dynamic binary translators
- Binary optimizers
- Language VMs

Process VMs

- Digital FX!32: Emulator used in Windows NT to support Alpha processors

Process VMs

- High-level language VMs
  - compilation
  - loader
  - interpreter/compiler



System VMs

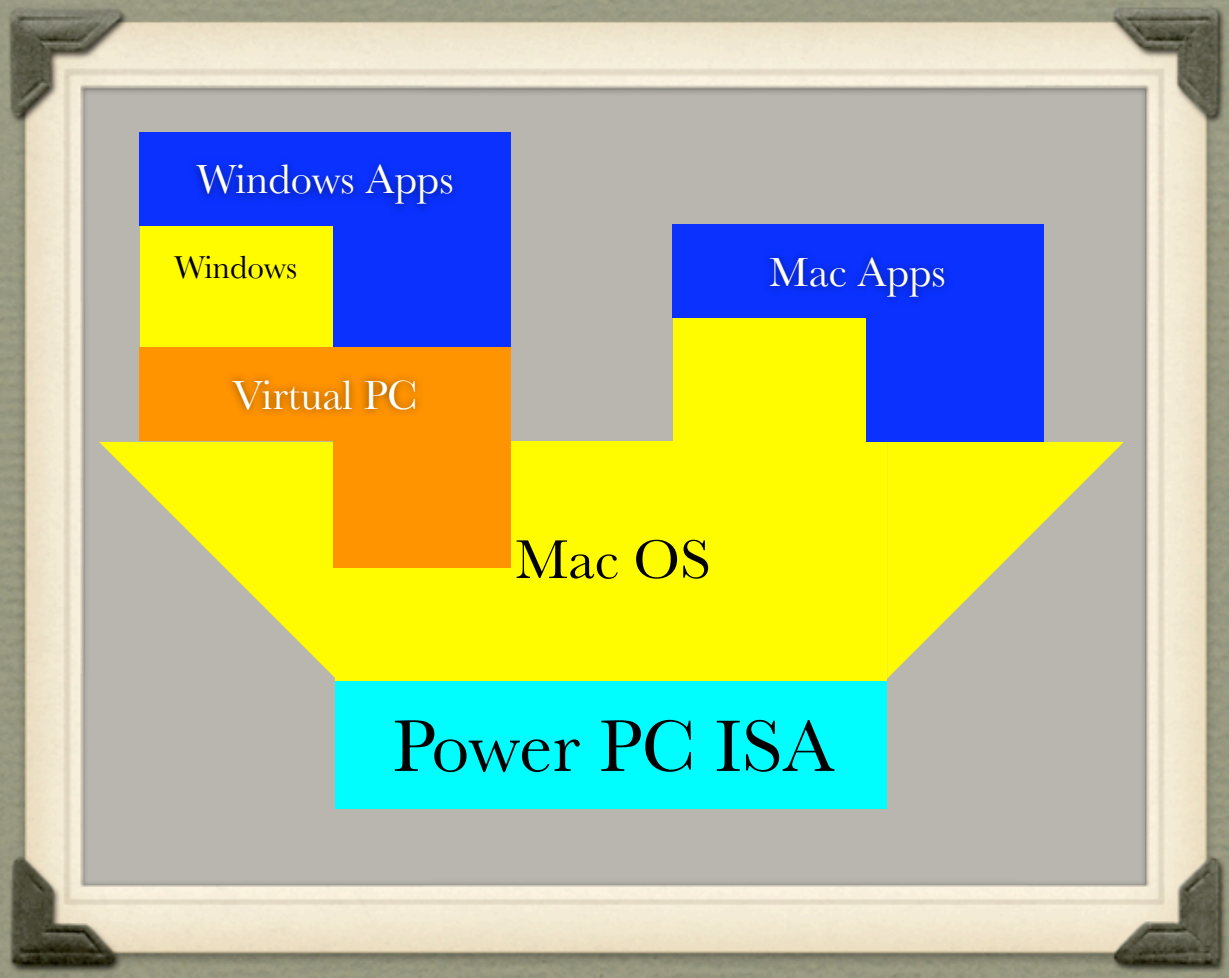
- VM Architecture (Popek and Goldberg, 1974)
  - VMM is placed on bare hardware with administrative privilege
    - transparently intercepts and implements OS's actions
  - Guest systems run with lesser privileges

System VMs

- Hosted VMs
  - virtualizing software on top of an existing host OS
  - typically rely on the same ISA
    - can use drivers and low level services provided by the host system
      - VMware

Systems VM

- Whole system VMs
  - host and guest systems do not share ISA
    - virtualize all software through binary translation
      - guest system ISA operations -> OS calls made to the host OS



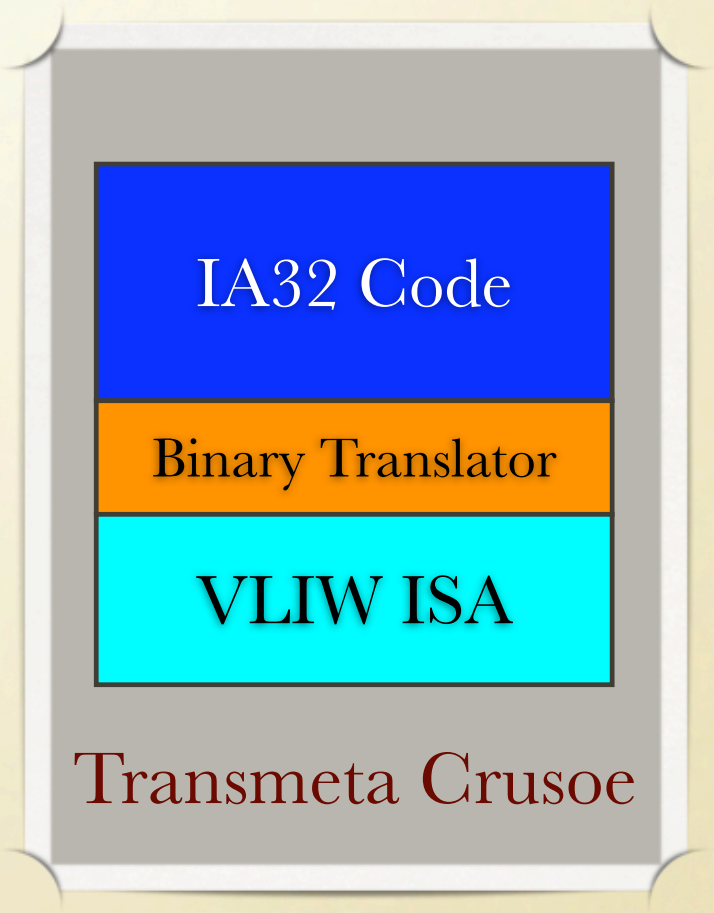
System VMs

System VMs

- Codesigned VMs
  - hardware or software translations
    - hardware: microOps
    - software: Transmeta Crusoe

System VMs

- Software translation
  - special “invisible” memory set at boot time
  - code in this memory execute in supervisor mode
  - translate and optimize guest instructions
    - cache these translated instructions



Summary

- Versatility of VMs
  - portability
  - high-performance
  - energy efficiency