Process as virtual machine (abstract)

Software virtual machine eg: JVM.

Alternative: complete hardware emulation.
So OS will run on virtual hardware.

Hardware virtual machine
- Efficient if it can directly execute most code.

Kernel space (emulated)

If simple program
- no privileged instruction
- not much I/O
- user code. So 99.9% code will execute directly.

Emulated execution of privileged instructions (supervisor)
- if supervisor instruction
  - then fault

Direct execution

Fault
- privileged instruction

Run instruction
- return to instruction that failed.

Ideally OS would shutdown such an instruction.
hypervisor maintains pageable so that guest OS thinks it has certain memory

* Tricky part of virtualization *

exception vectors of guest OS in "fake" location (so that HW does not access that vector)
So we need 2 levels of mapping

but HWU does only one level

(Guest OS uses shadow page tables)

"load CR3"
privileged

load CR3
real page table

I/O in virtual machine

Guest OS will have a "fake" h/w space in memory for memory mapped I/O

* There need not be anything in the memory. If OS tries to put byte in say frame buffer, then page fault
* VM will transfer to real frame buffer.

* Handling non virtualizable instruction *

1. Full emulation of kernel code.
2. Binary translation of JIT compiler.

```
guest instruction  emulated CPU

mov R1, R2  →
(Too many jumps!)

Load R1, address
Store R2, address
(Translated to longer
  simple code)
```

Done in JVM, Mozilla's respected JavaScript execution

> Alternate method: Para virtualization

Guest OS.

Register syscall handler:
```
pagefault
```
Update VM mappings.

System call API's are provided within VM. So no "fake system calls"
Advantages/Disadvantages of Paravirtualization:

- Using interfaces that have been optimized.
- Can't boot on unmodded OS. Currently, we have Zen.

* Another option: "hardware virtualization"
  - We make ISA have different level like:
    1. Supervisor
  - So we will guest OS exception handler of virtual machine.

  - Difficult to implement hypervisor on this model.