What needs to be virtualized?

Exactly what you would expect

- CPU
- Events (exceptions and interrupts)
- Memory
- I/O devices

Isn't this just duplicating OS functionality in a VMM?

- (yes) approaches will be similar to what we do with OSes
  simpler in functionality, though (VMM much smaller than OS)
- (and no) but implements a different abstraction
  hardware interface vs. OS interface

Approach 1 : complete machine simulation

➡ Simplest VMM approach, used by *bochs*

Build a simulation of all the hardware

- CPU – a loop that fetches each instruction, decodes it, simulates its effect on the machine state (no direct execution)
- Memory – physical memory is just an array, simulate the MMU on all memory accesses
- I/O – simulate I/O devices, programmed I/O, DMA, interrupts

◉ Too slow!

- CPU/Memory – **100x slowdown**
- I/O Device – 2x slowdown

➡ Need faster ways of emulating CPU/MMU