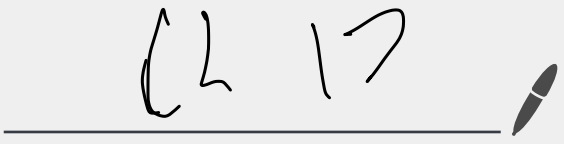
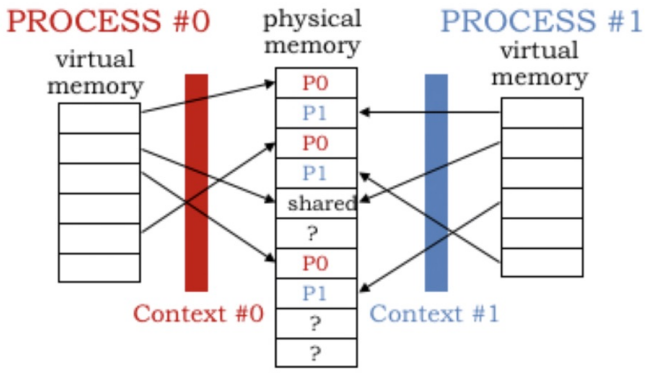


Building a Virtual Machine (VM)



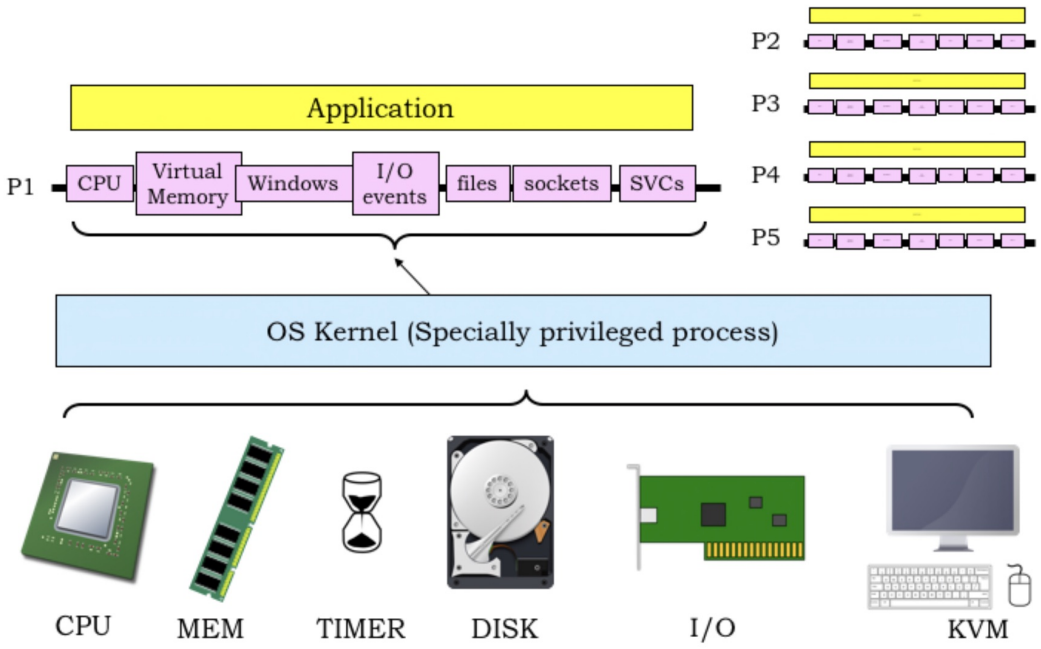
Goal: give each program its own "VIRTUAL MACHINE"; programs don't "know" about each other…

New abstraction: a process which has its own

- machine state: R0, …, R30
- context (virtual address space)
- PC, stack
- program (w/ shared code)
- virtual I/O devices

"OS Kernel" is a special, privileged process running in its own context. It manages the execution of other processes and handles real I/O devices, emulating virtual I/O devices for each process.

One VM For Each Process



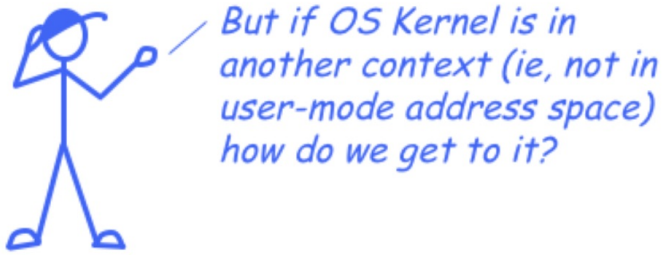
Communicating with the OS

User-mode programs need to communicate with OS code:

Access virtual I/O devices

Communicate with other processes

…



Solution:

Abstraction: a supervisor call (SVC) with args in registers – result in R0 or maybe user-mode memory

Implementation:

use *illegal instructions* to cause an exception -- OS code will recognize these particular illegal instructions as a user-mode SVCs