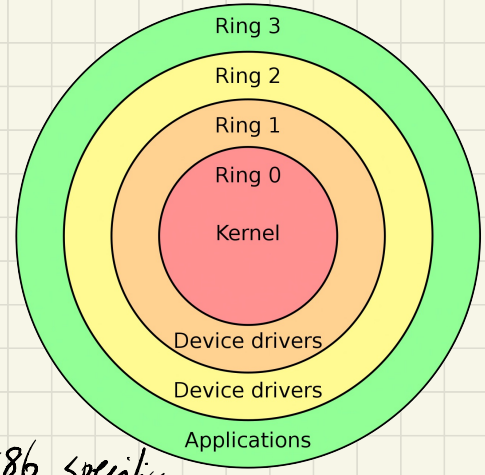
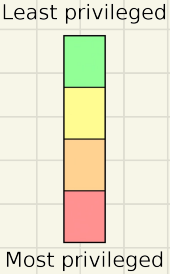
9/29/23 Isolation

Key role of OS: abstract & manage hardware
→ processes must access hw via OS services
→ need privilege levels

Protection Rings



x86 specific.



register %cs

mode bit

☆ Ring 0 (kernel mode)
→ access to privileged instr.
→ eg. I/O port, halt, update virtual memory mapping, disable interrupts
→ access to all mapped virtual memory

☆ Ring 3 (user mode)
→ only nonprivileged instr.
→ eg. add, push, request for mode switch
→ only user accessible virtual memory

☆ ring 1 & 2 does not have access to privileged instr.

Types of Mode Transfer

→ system calls.
- → Kernel service APIs
- → syscall, sysret instr.
- → requested by user! (synchronous)
- → resume on next instr. on return

→ interrupts
- → hardware notifications
  - → I/O completion (disk write, packet arrival), timer interrupt
  - → unrelated to the current instr. (asynchronous)
  - → resumes on the interrupted instr. on return

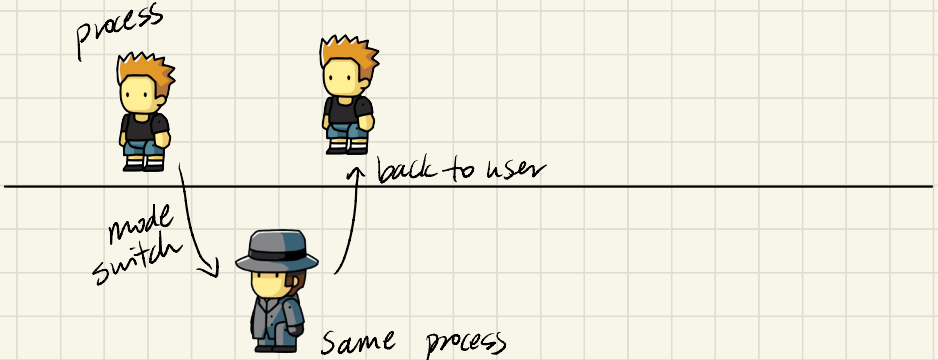
★ needs to be handled in a timely fashion

→ Exceptions
- → unexpected problem on current instr. (synchronous)
  - → access invalid memory (nullptr, segfault), divide by zero, execute privileged instr.
- → terminate process, or handle the exception and resumes (retries the faulting instr.)

- Execution Model
  - → process
    - → runs arbitrary logic in user mode
    - → runs fixed kernel code in kernel mode

on mode switch
states
- → different code (PC)
- → different stack
- → different register values

☆ must save process's states in order to resume execution after switching back to user mode



user mode

kernel mode

☆ syscall = voluntary transition

☆ exception = unintended problem

☆ interrupt = urgent task requiring