Operating system (kernel)

- Make it possible for user/application to share resources (hardware)

- IS a multiplexer

\[ P_1, P_2, P_3, P_4, P_5 \]

Many processors share same resources

Share -> CPU, I/O, disk, memory and others

Key idea in OS: Virtualization

Each app. and each user thinks they are running alone
Specific things OS virtualizes
- CPU - all processes get a turn to run
  - "Scheduling"
- I/O - syscalls to request the OS do I/O on your behalf
  - read/write/ioctl
- Disk - filesystems - open file table - vnode table
- Interrupts - forwards interrupts to processes
- Memory - focus after test
  - Each process thinks it has access to all memory
  - How to virtualize this memory?

Each program thinks it is alone
⇒ OS provides Isolation
Context Switches -> Multiplexing the CPU

How?
- interrupt! timer interrupt

- Store all info from P1 in memory
- re-store all of P2 info from memory
- "architectural state"
  - registers
  - flags
  - etc.

How often? -> Scheduling quantum
5s 10ms -> 100ms

depends on store/restore... ~10ms

(-create new process -> fork())