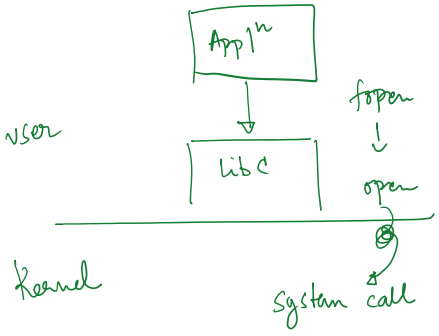
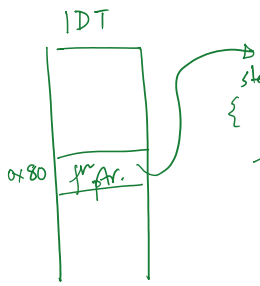
Lecture 12
09 August 2018 11:35

Lecture 12. 9th Aug. 2018

① system calls

int 0x80 | eax

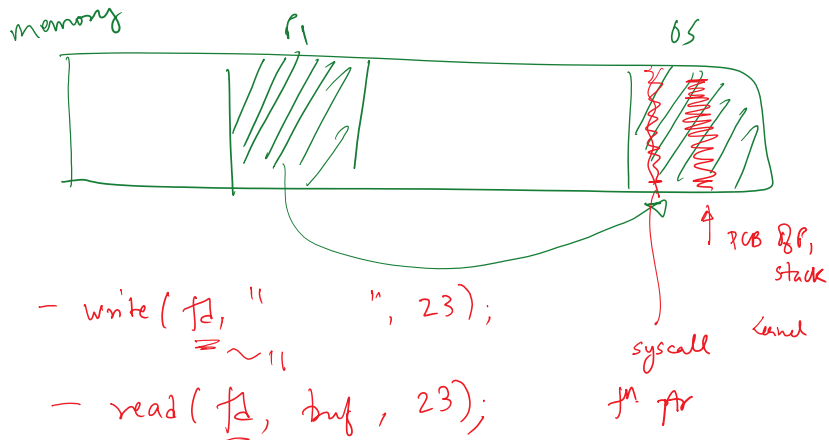




stemcall_handler
{
fn = syscall table [eax];
call fn;

* process context for system calls.

– "system calls execute in a process context". | interrupt context



– write(fd, " ", 23);
~ "

– read(fd, buf, 23);
[0x abcdef

② Recap.

– Goal of an OS is to provide abstractions (for general purpose computing) to save the world! — need resources!

– tasks — resources mgmt. — isolation / protection

– transparent / interface-driven
– efficiency!
} perf. / execution

* two building blocks (
modes of execution / limited direct execution || assisted

(*) two enabling blocks (
- modes of execution / limited direct execution || assisted
- interrupt-driven execution.

⇒ principle: let user programs run directly on the CPU as far as possible.

- only intervene when required.
  - bad stuff
  - interrupts

⇒ OS still paranoid about regaining control of the CPU.

(3) process
- execution entity
  - memory, files, variables etc.
  - implicit assumption of CPU(s).

process view

---

abstraction

virtualization / multiplexing

hardware / resource