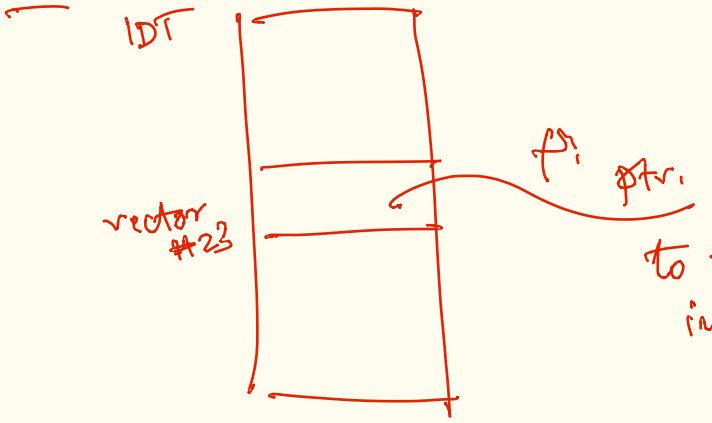
Lecture #8 CS347 || system calls

xv6 introduction

- recap:

ABI, API, requirements of the system call mechanism

int instruction, IDT, IDTR

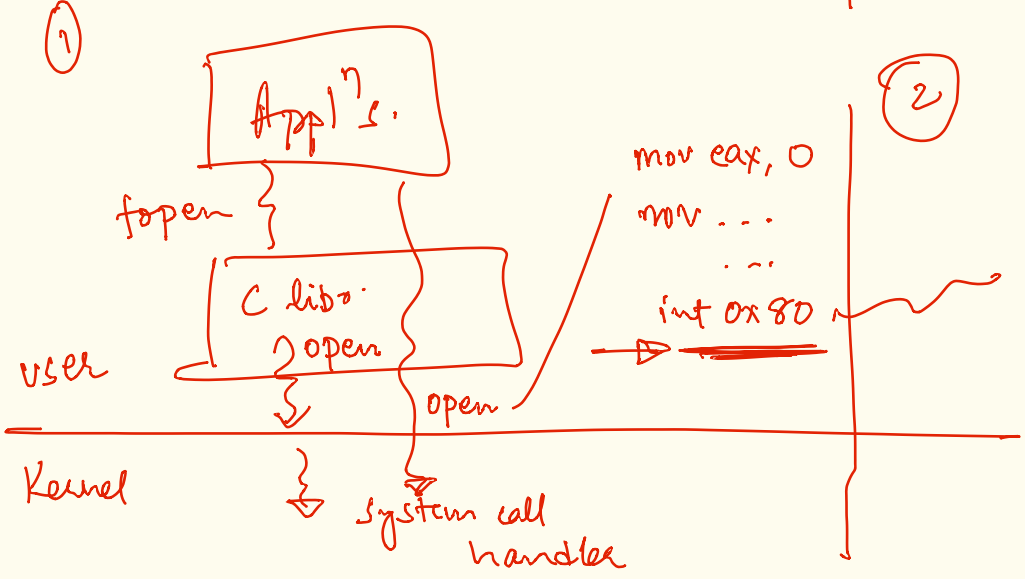


system call

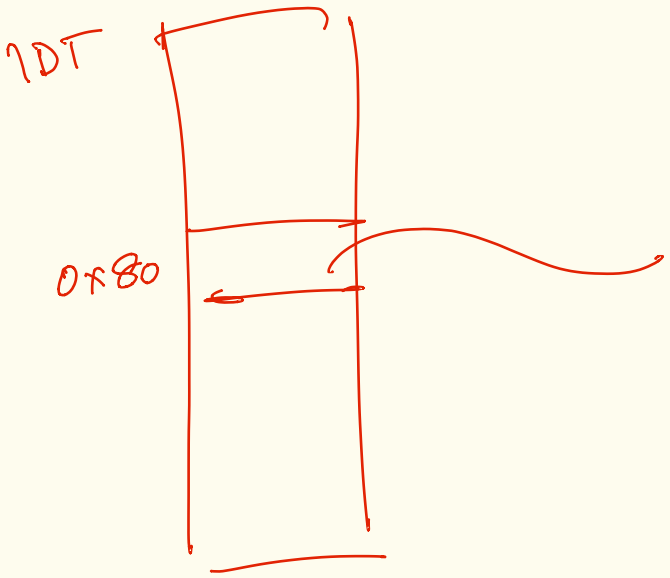
⇒ explicit software interrupt.

system call identification & arguments.

- via CPU registers ~ eax ← store the system call number (identifier)
  ebx
  ecx
  ⋮
- via memory / user stack ⟩ SP



④ how to get to THE system call.



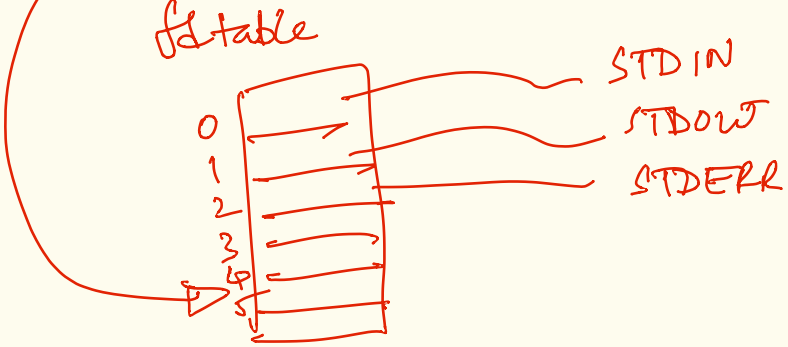
P_1 P_{23}

```
syscall_handler () {
    fn = syscall_table[eax];
    call fn;
    iret;
}
```

scheduler();

P_1 on reschedule

```
int fd = open(          );
```



xv6 introduction ++

- teaching OS MIT
- xv6 the source code ⇒ implements the OS logic
  ⇒ generates the OS program binary
  ⇒ xv6.img (1)
- qemu — machine emulator
  - xv6.img — Kernel / OS to boot the machine into
  - fs.img — root/default file system / files available on boot up.
- makefile
  - xv6.img
  - fs.img
  - qemu