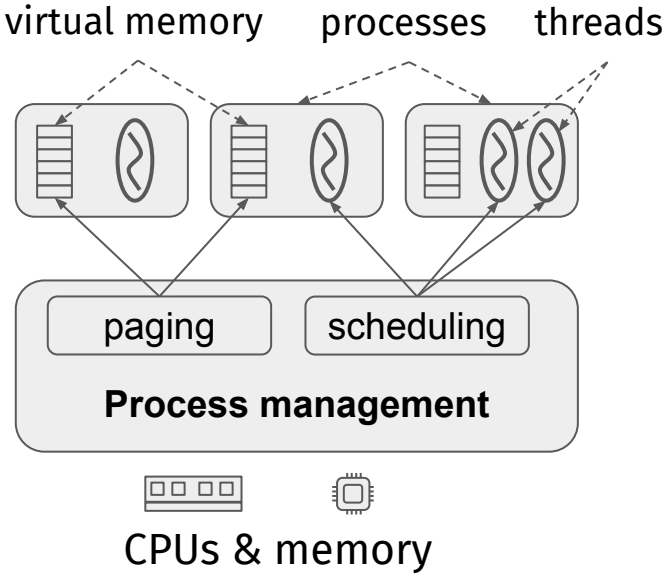
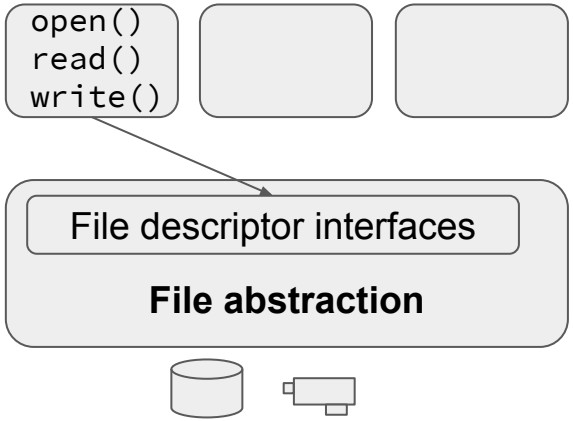
Kernel Space Programming

Michio Honda

Roles of Operating Systems

- **Hardware abstraction**



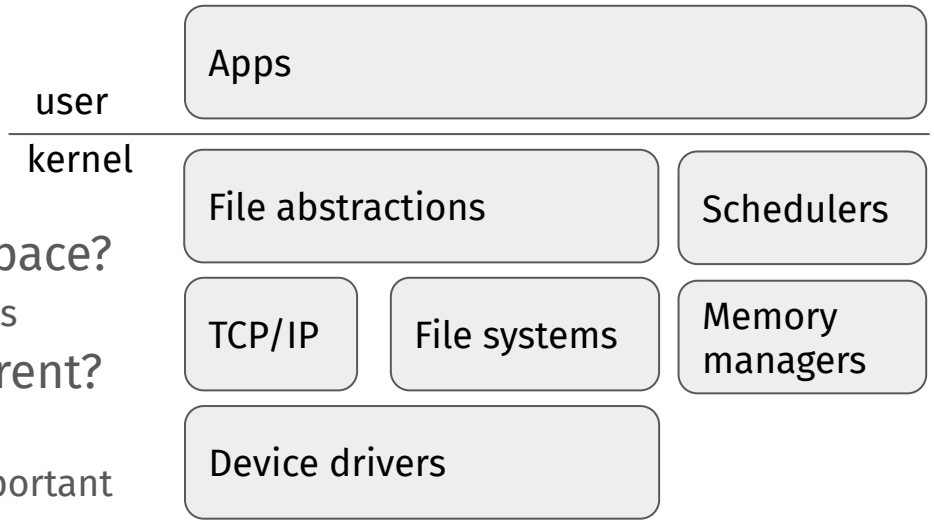


`ps, top, vmstat, /proc/*/fd etc.`

`/proc/*/fd, /proc/cpuinfo, lspci, etc.`

User Space and Kernel Space

- User space
  - Protected (crash only affects itself)
  - Limited resource access
- Kernel space
  - Unprotected
  - Access to everything
- When do we code in the kernel space?
  - Extending/adding kernel components
- How is kernel programming different?
  - Crash affects the entire system
  - Concurrency control is extremely important
    - multiple readers and single writer
    - sleepable or non-sleepable
    - etc



Hand-on (kernel module & memory allocation)

(in your VM)

```
git clone https://github.com/micchie/uoe_sysprog.git
cd uoe_sysprog/week5
cp hello-kmalloc.c hello.c
make
sudo insmod ./hello.ko
sudo rmmod ./hello.ko
sudo dmesg | tail
```

Hand-on (locking)

(in your VM)

```
cp hello-lock.c hello.c
make
sudo insmod ./hello.ko
sudo rmmod ./hello.ko
sudo dmesg | tail
```

Hand-on (file operations)

(in your VM)

```
cp hello-fops.c hello.c
make
gcc hello-test.c
sudo insmod ./hello.ko
sudo ./a.out
sudo rmmod ./hello.ko
sudo dmesg | tail
```

(edit hello-fops.c to print something in hello_read() and hello_write(), then repeat above steps)

This is where the OS's file abstraction comes in - you can implement your own kernel feature so that application can use the common file operations (e.g., read() and write()) for it!

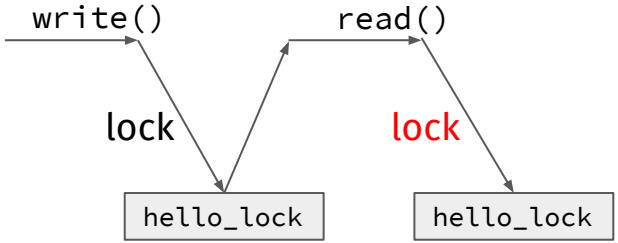
Hand-on (dead lock)

(in your VM)

(edit hello-fops.c to remove the spin_unlock() in hello_write())

```
cp hello-fops.c hello.c
make
gcc hello-test.c
sudo insmod ./hello.ko
sudo ./a.out
```

(your system will freeze)



Kernel space objects can be accessed by any, multiple processes (and interrupts and other kernel threads). Concurrency control is very important.

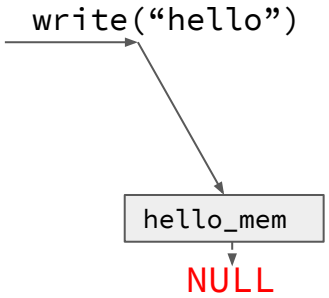
Hand-on (kernel crash)

(in your VM)

(edit hello-fops.c to remove the kmalloc() in hello_init())

```
cp hello-fops.c hello.c
make
gcc hello-test.c
sudo insmod ./hello.ko
sudo ./a.out
```

(your system will crash)



NULL pointer dereference immediately crash the entire system.