Operating Systems (CS3000)

Lecture – 8

(System Calls and OS Structure)

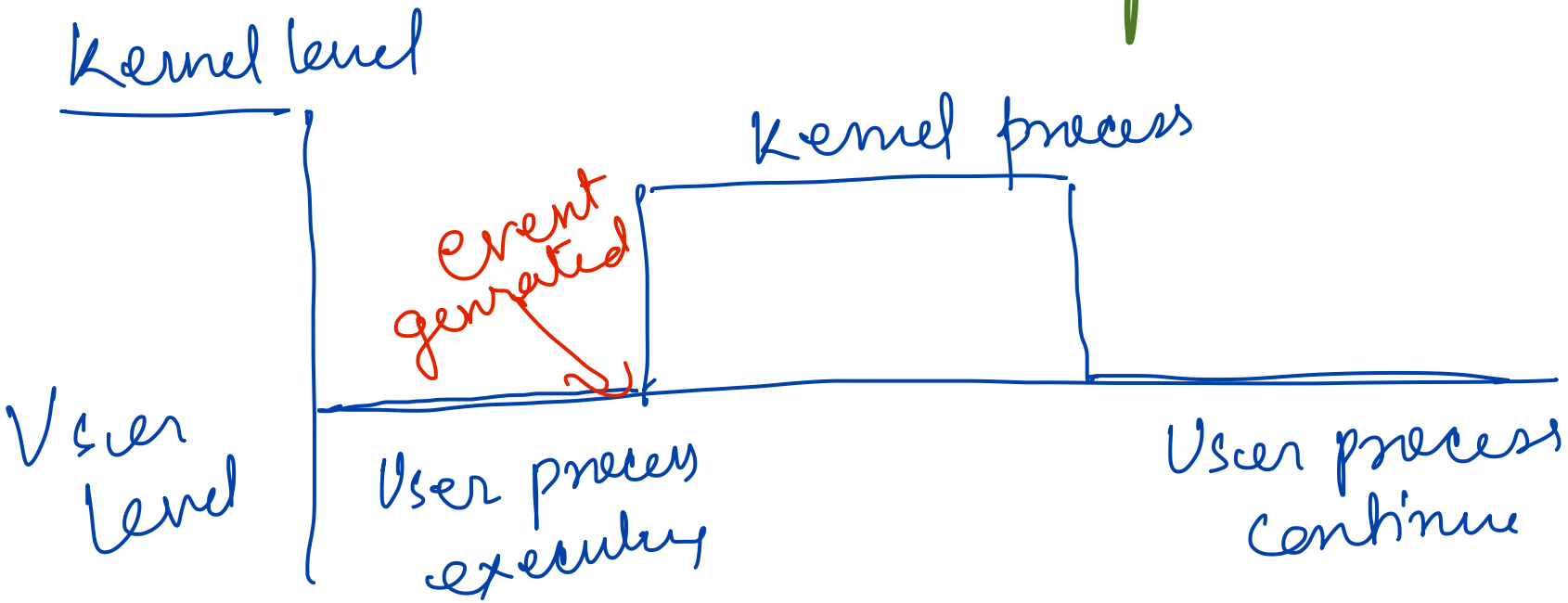
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

Dr. Jaishree Mayank

Assistant Professor

Department of Computer Sc. and Engg.

* OS is event-driven program
— executes only when there is an interrupt



Events

Hardware Interrupts:

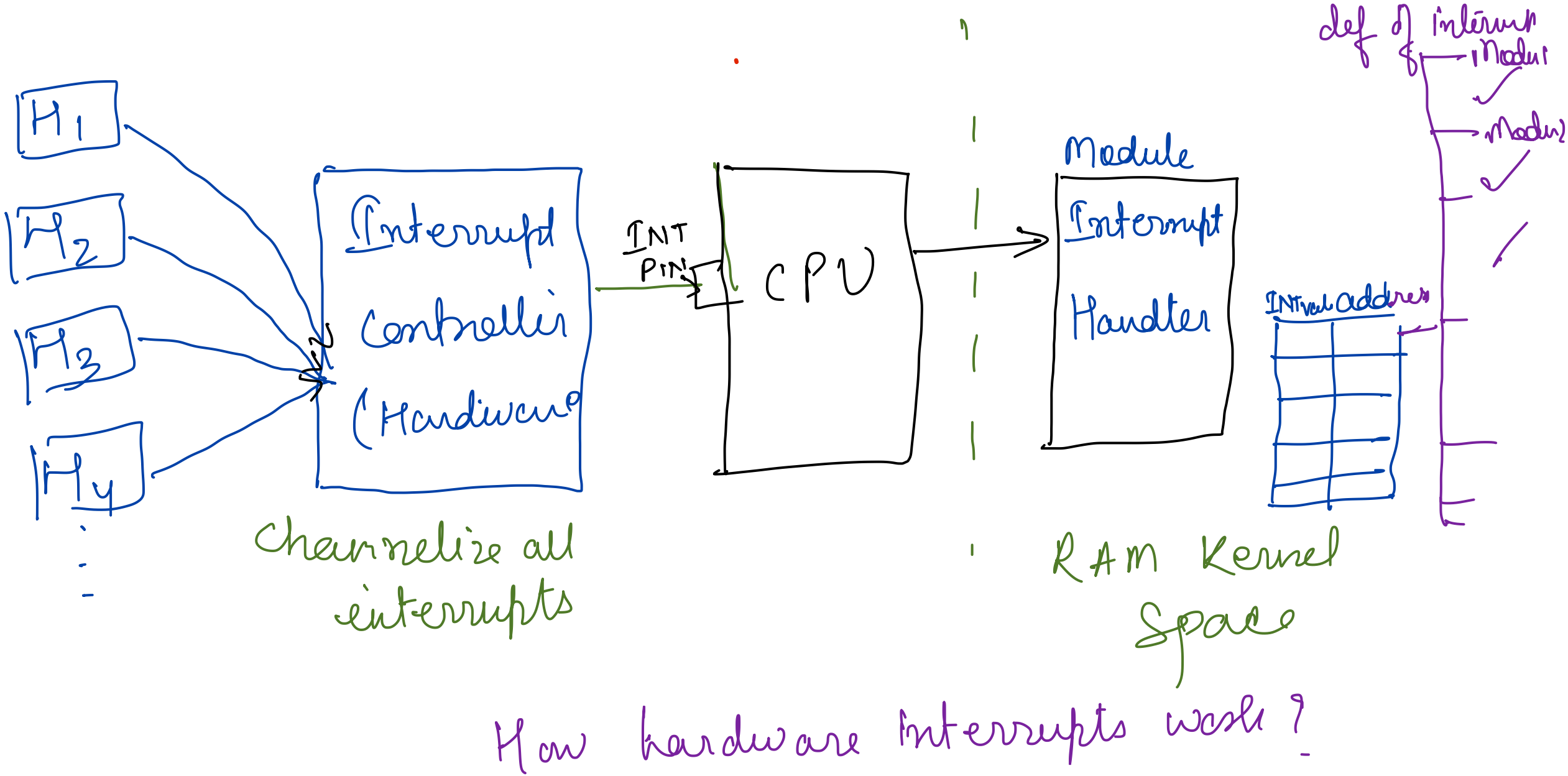
- Raised by any hardware device
- They are asynchronous and may occur any time

Traps:

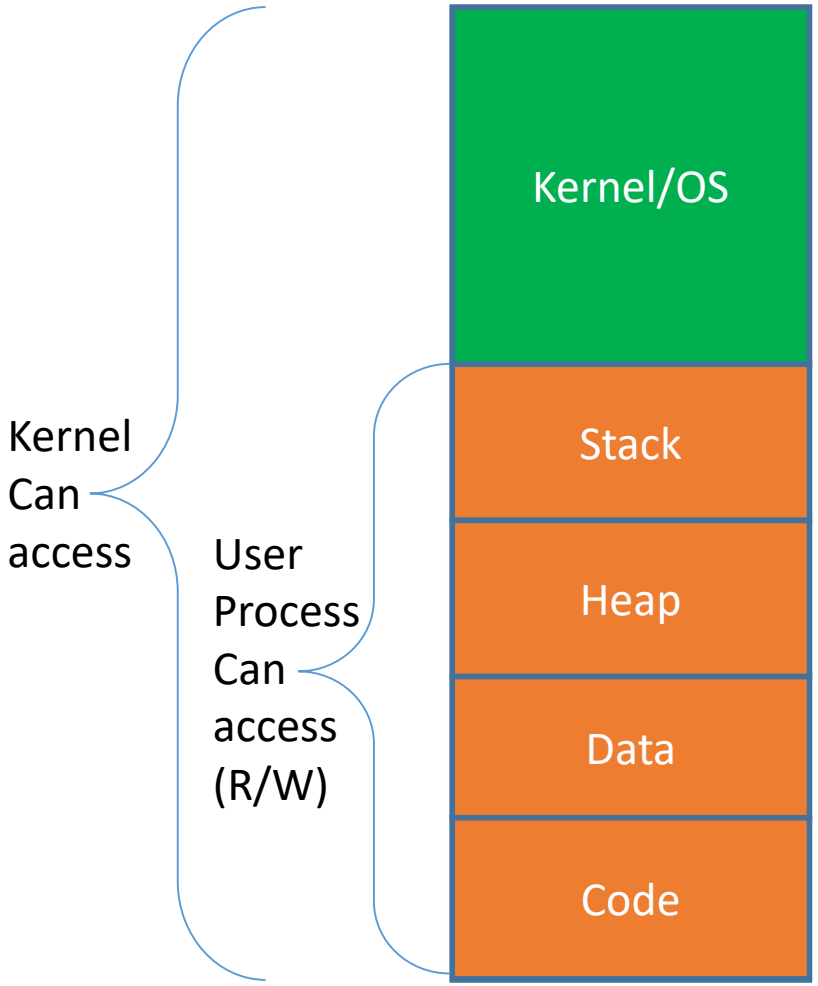
- Sometimes called software interrupts
- Raised by user program, to access OS functionality

Exception:

- Generated automatically by the processor itself due to illegal instructions
- Faults and Aborts


H1
H2
H3
H4
Interrupt
Controller
(Hardware)
INT
PIN
CPU
Module
Interrupt
Handler
def of interrupt
Module1
Module2
INTR vector address
Channelize all interrupts
RAM Kernel Space
How hardware interrupts work?

Process Memory Map



Memory Map of Process

1 – Instructions of OS
2 – OS data

* User process can access only user space

* Kernel process can access both user and kernel space

How do User Program invoke OS?

?

Solution :— System Call.

Communicating with the OS (System Calls)

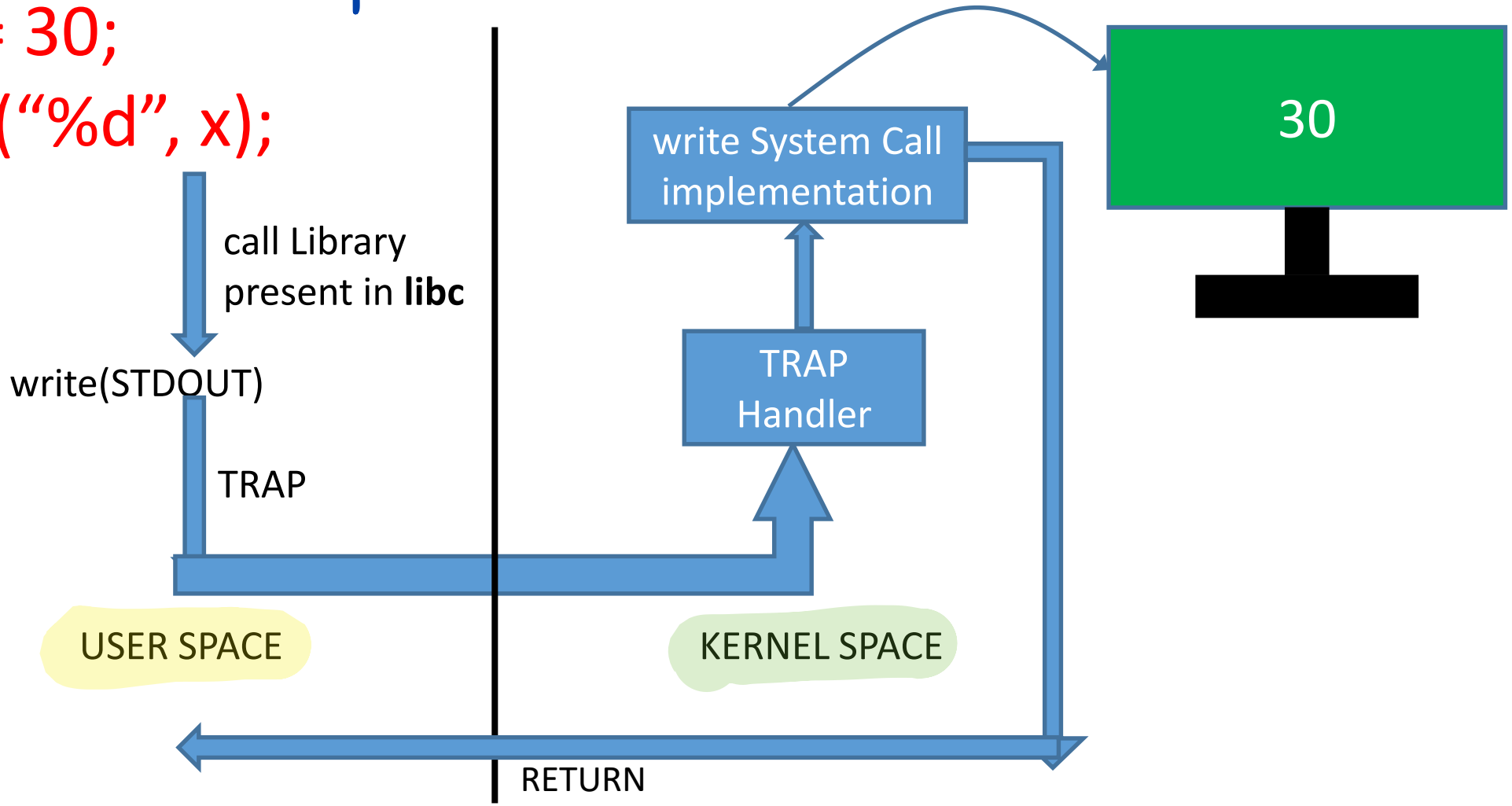
- System Calls are a set of special functions which the OS supports.

- User Process can invoke any of the system calls

- Why?
  - to get information
  - to access hardware/ resources within the Kernel.

Example of Software Interrupt printf( )

```
int x = 30;
printf("%d", x);
```



What Happens During System Calls?

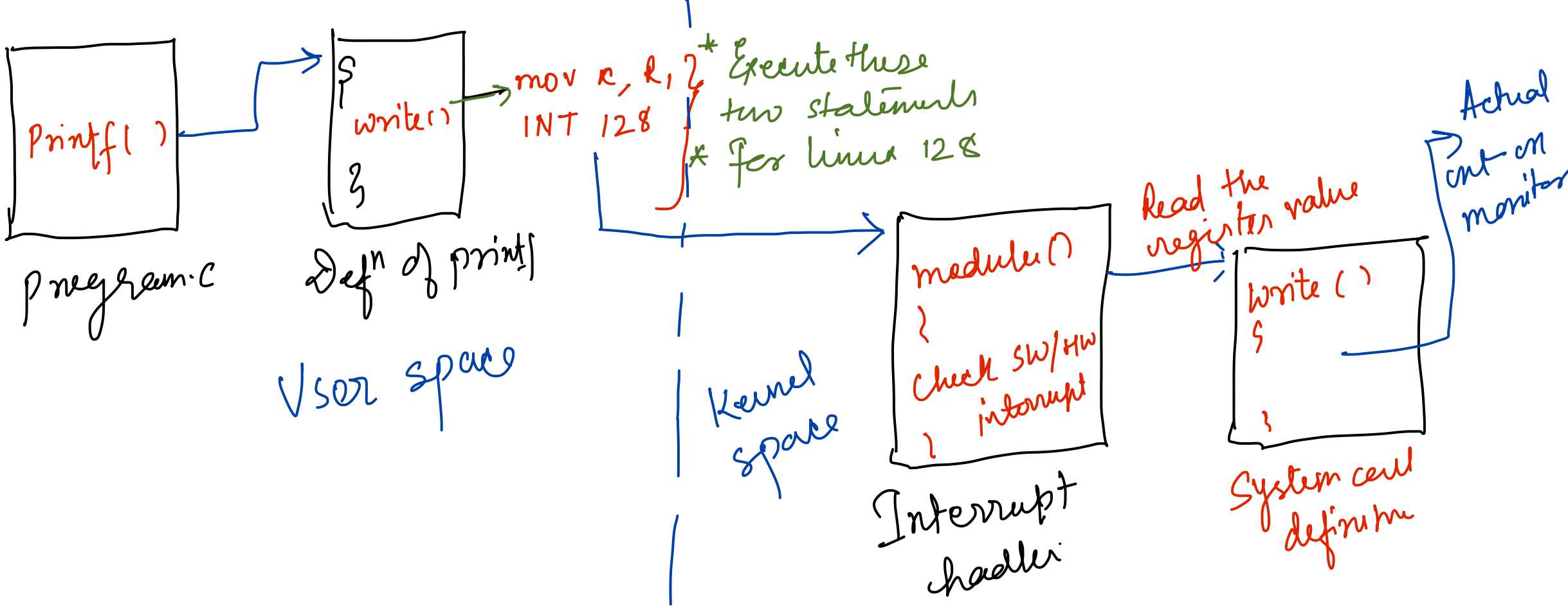
- process (user mode) → process (kernel mode)
  - allow the kernel or the operating system to actually execute the task

- System Call completes: process (kernel mode) → process (user mode)

Function Call v/s System Call

- CALL instruction
- User Space
- CALL jumps to a relocatable address

- TRAP instruction
- User Space → Kernel Space
- TRAP jumps to a fixed address

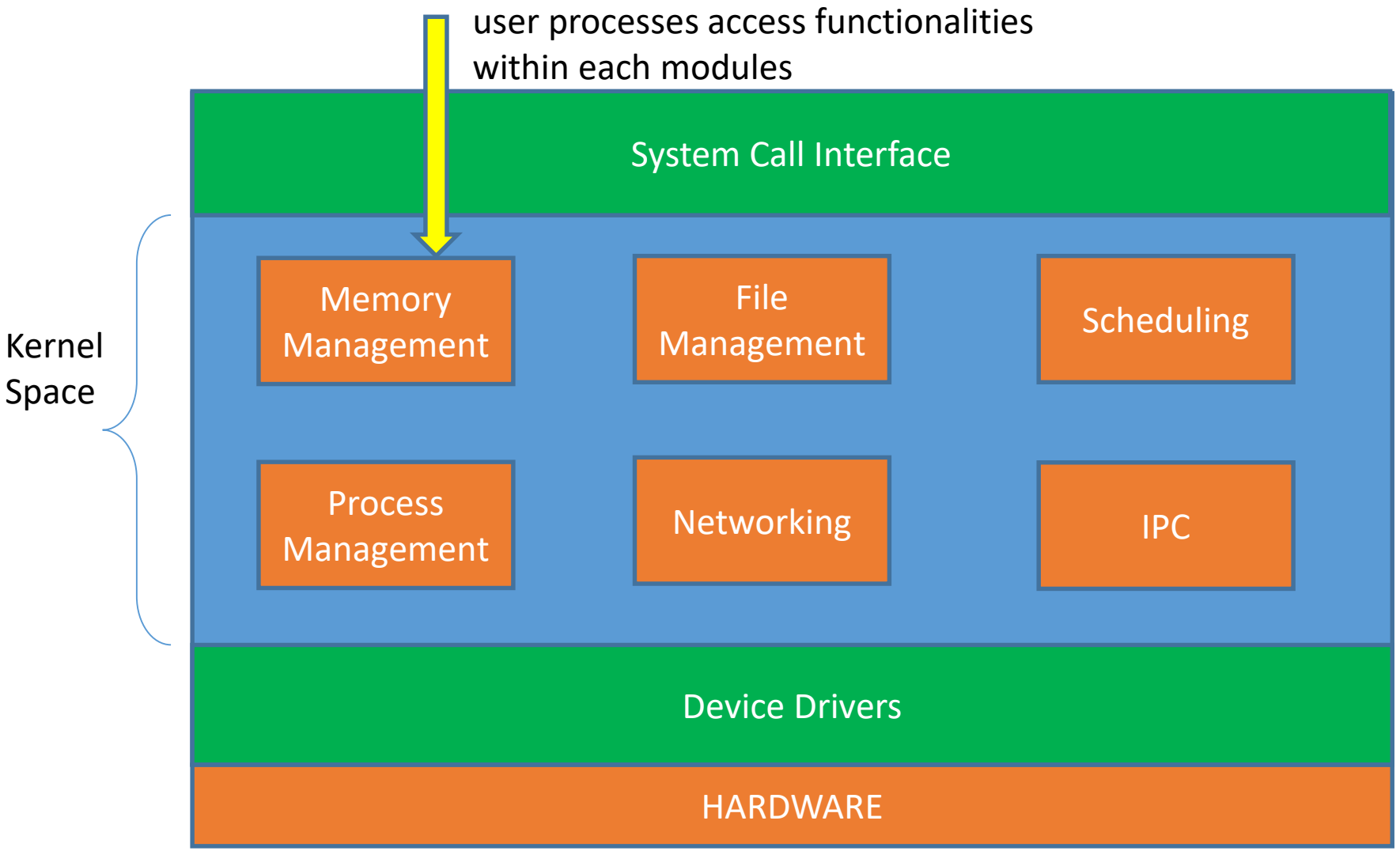


How Software Interrupt Works?

System Calls for Files

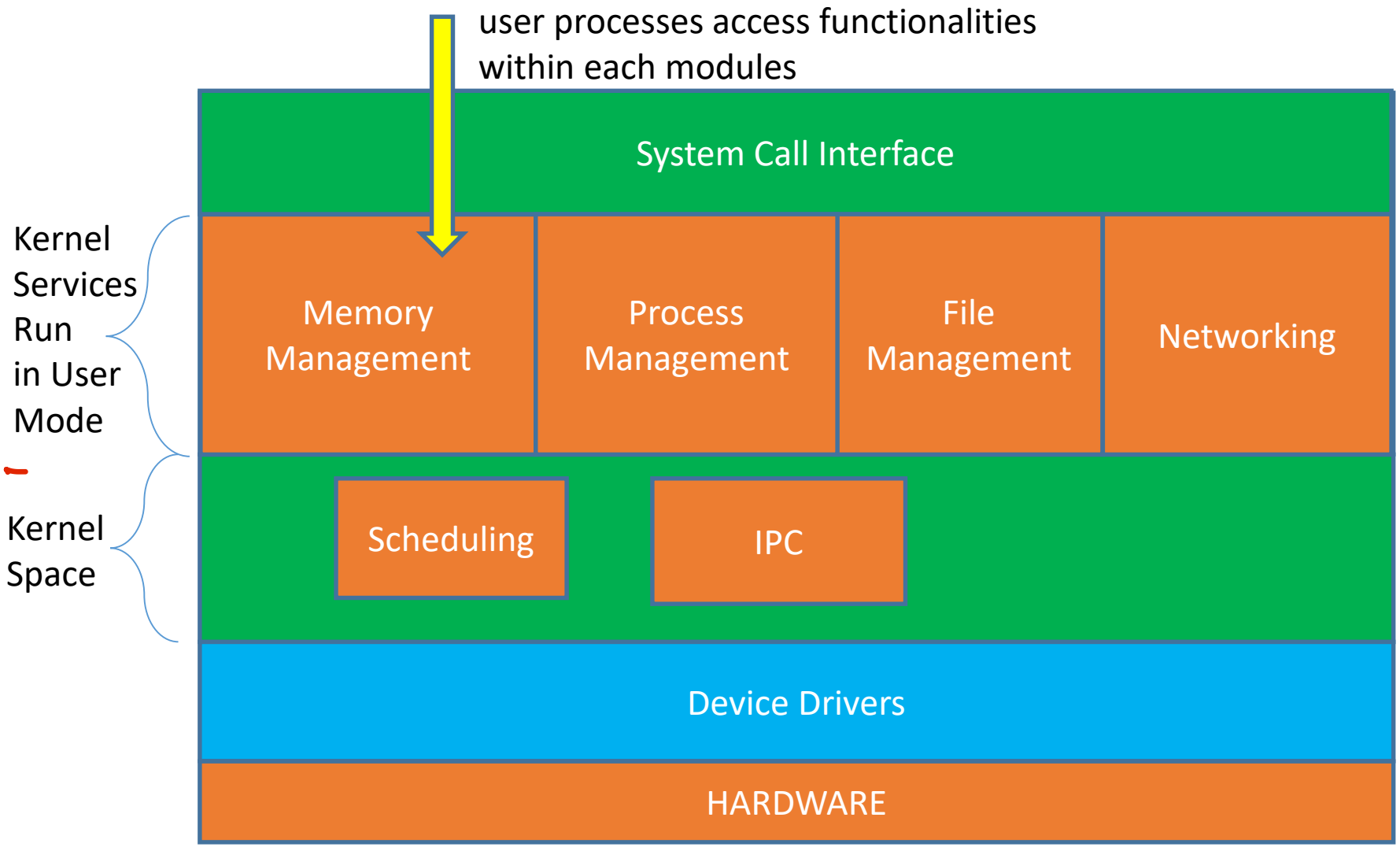
- Files: Data remains even if power is off
- Operations on Files:
  - open()
  - close()
  - read()
  - write()
- Files are stored in HDD. System call required to access Hardware.
- Process -> System Call -> Kernel -> HDD -> Return FP to Process

Monolithic OS Structure



1. All Components of OS/All functionality of OS are present in Kernel Space
2. Kernel is a single process where all functionalities share the same address space.
3. Direct Function Calls between modules of Kernels
4. Large Size Kernel, Difficult to maintain and manage

Microlithic OS Structure

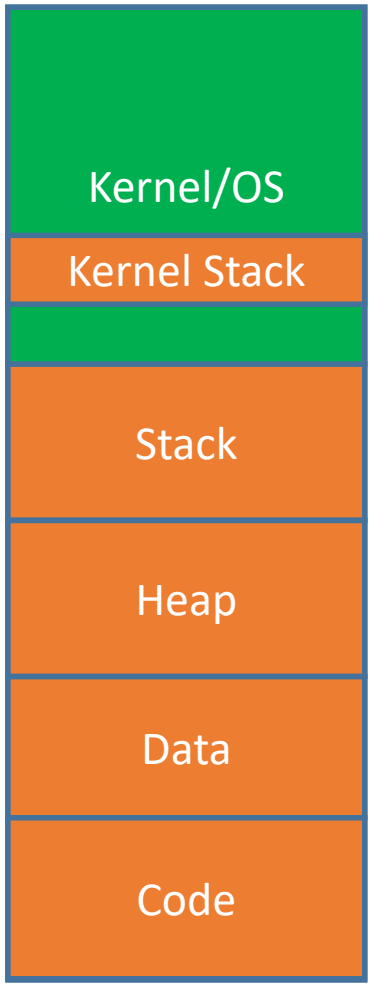


1. Few Components of OS/few functionality of OS are present in Kernel Space
2. Some Kernel services can be modified by users based on needs.

1. Direct Function Calls not possible. IPC is required.
2. Small Size Kernel, Easy to maintain and manage

What Metadata of a Process Kernel Stores?

- PCB
- Kernel Stack for User Process
  - During System Calls
- Page Table for that User Process



Thank You

Any Questions?