CSCI 3753: Operating Systems Fall 2024

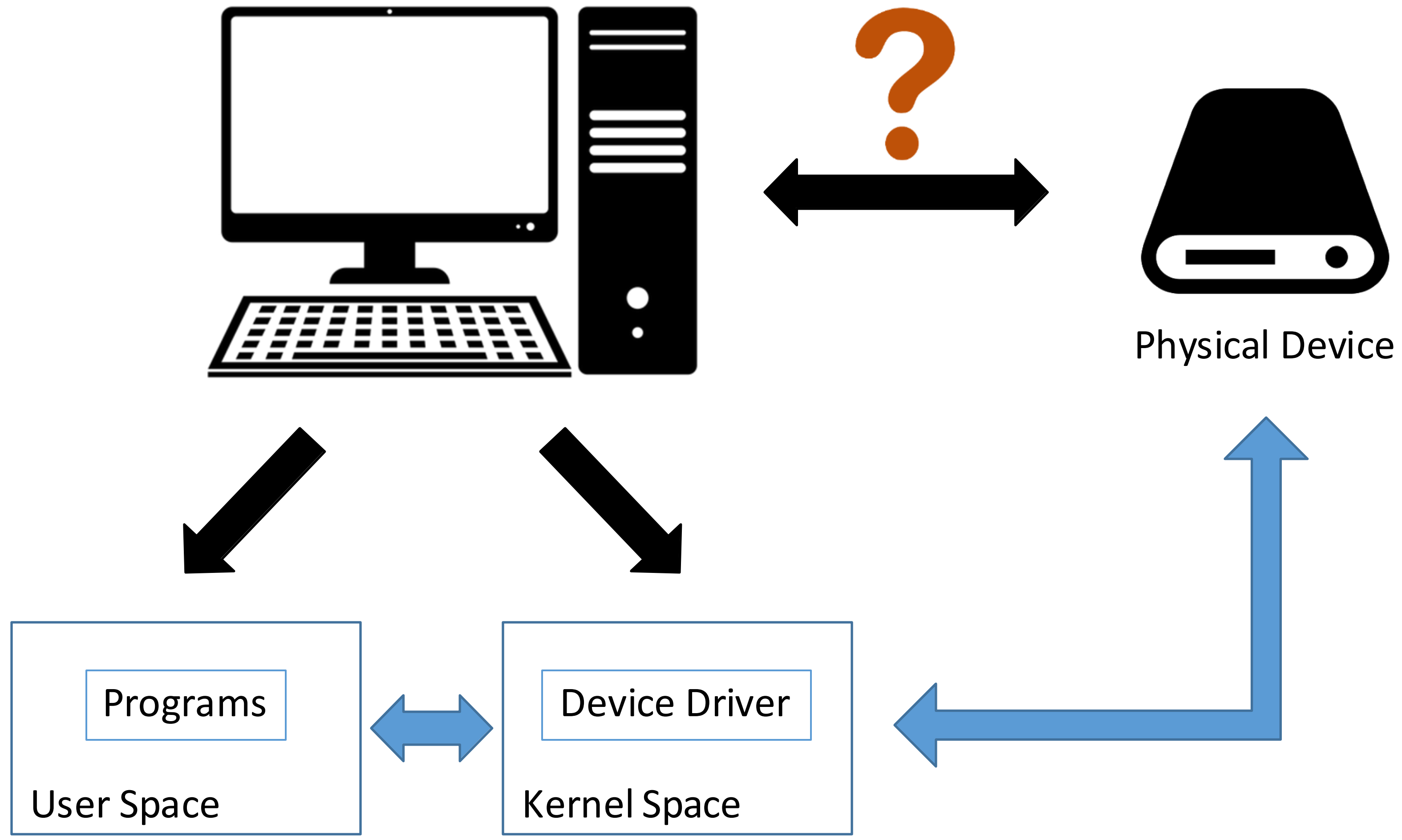
Dylan Sain

Department of Computer Science

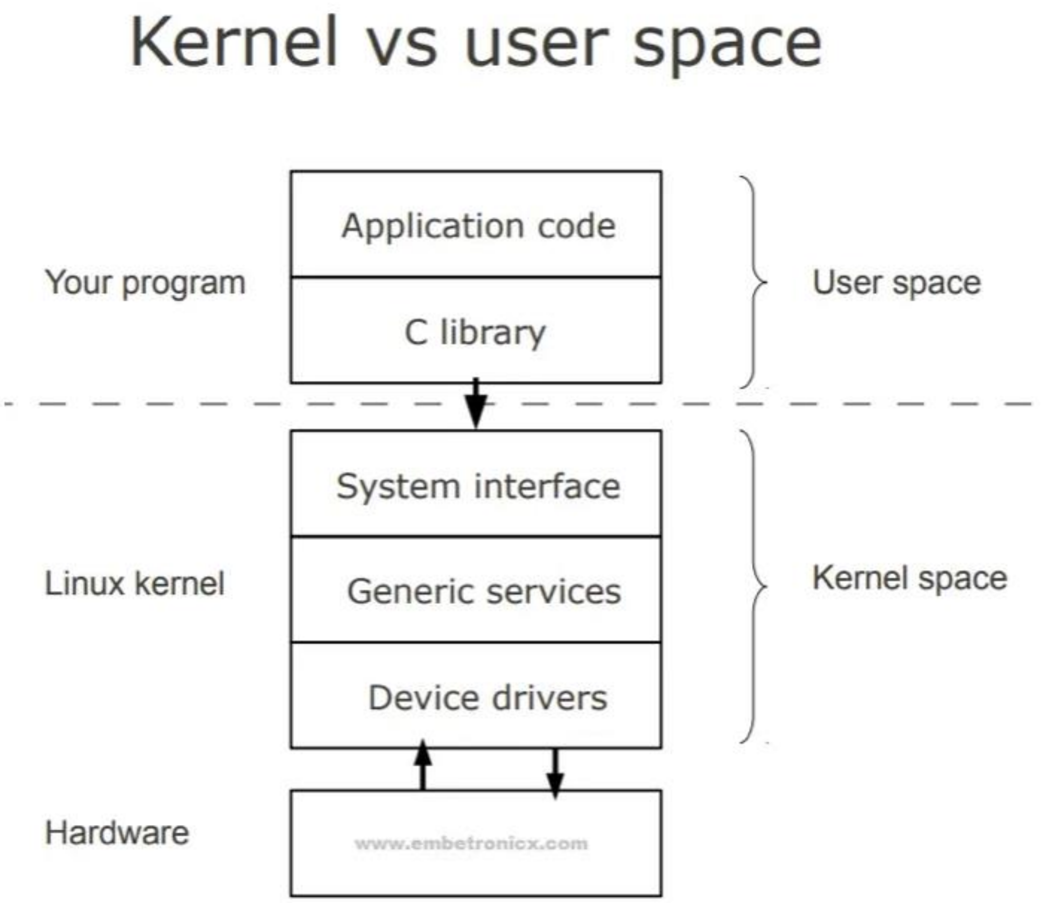
University of Colorado Boulder

Week 4: PA2, PA3, and Device Drivers

An Overview


?
Physical Device
Programs
User Space
Device Driver
Kernel Space

Device Drivers



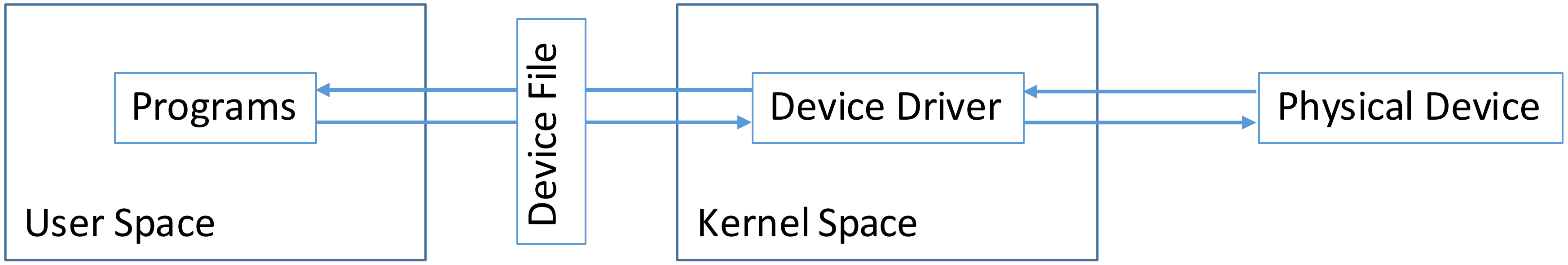
- A way for devices to interact with the OS
- Utilize LKMs to provide functionality as soon as they are plugged in

Types of Device Drivers

- Character Devices
  - One character at a time
  - Faster and smaller bits of information
  - Used for data streams
  - Mouse, keyboard, sound devices, ect.
- Block Devices,
  - Transfer *blocks* at a time
  - Size depends on the device
  - Commonly used for storage
  - USB drives, SSD, Hard Drives, ect.

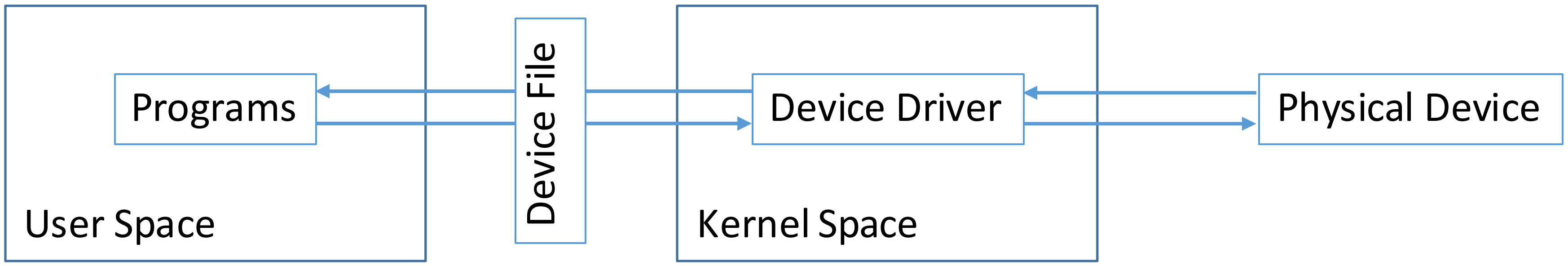
Device Driver

- Transfers data to and from a user process
- When a calling program invokes a routine in the driver, the driver issues commands to the device.
- Once the device sends data back to the driver, the driver may invoke routines in the original calling program.
- A program cannot access the driver in the kernel directly.



Why File I/O?

- When a device is plugged in the kernel creates a device file
- All operations to and from the device are done through the device file
- Unique to each device
- Makes it easier for the software to interact with the hardware



Activity: Explore the devices connected to your computer!

- Only Linux based devices (Sorry Windows!)
- Open a terminal
- Type "ls -l /dev"
- Try and see if you can identify some of the devices there!

PA2 and PA3

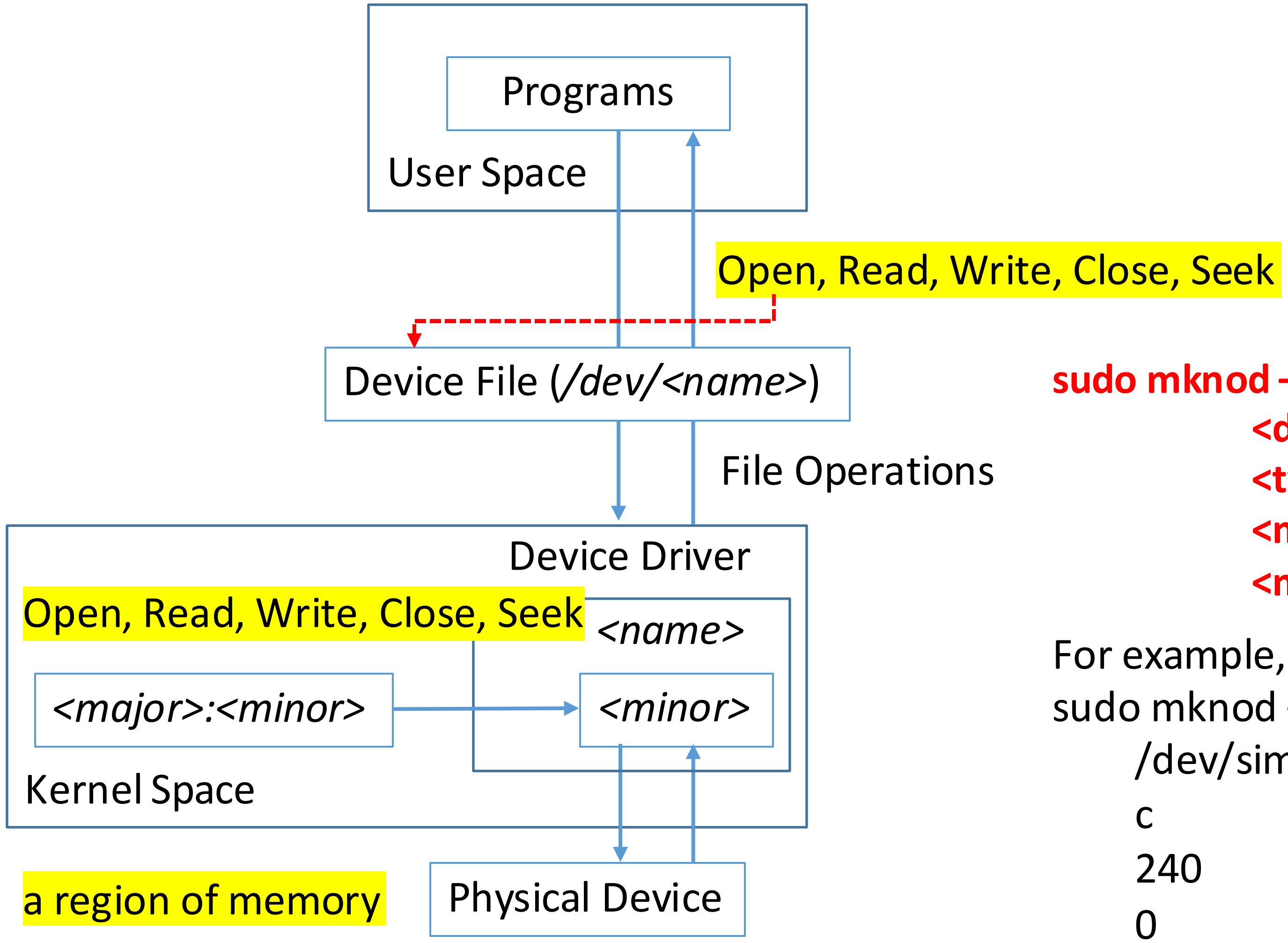
- PA2
  - Working with File I/O
  - Simple reading, writing, seeking

- PA3
  - Create your own device driver
  - Utilize skills made in PA1 (LKMs) and PA2 (File I/O)
  - Create and edit a device file
  - Create your own read/write/seek functions

Programming Assignment 2

Q & A

PA3 – Character Device File



sudo mknod –m <permission>
<device_file_location>
<type of driver>
<major number>
<minor number>

For example,

```
sudo mknod –m 777
    /dev/simple_character_device
    c
    240
    0
```

PA3 – Requirements

pa2_char_driver.c

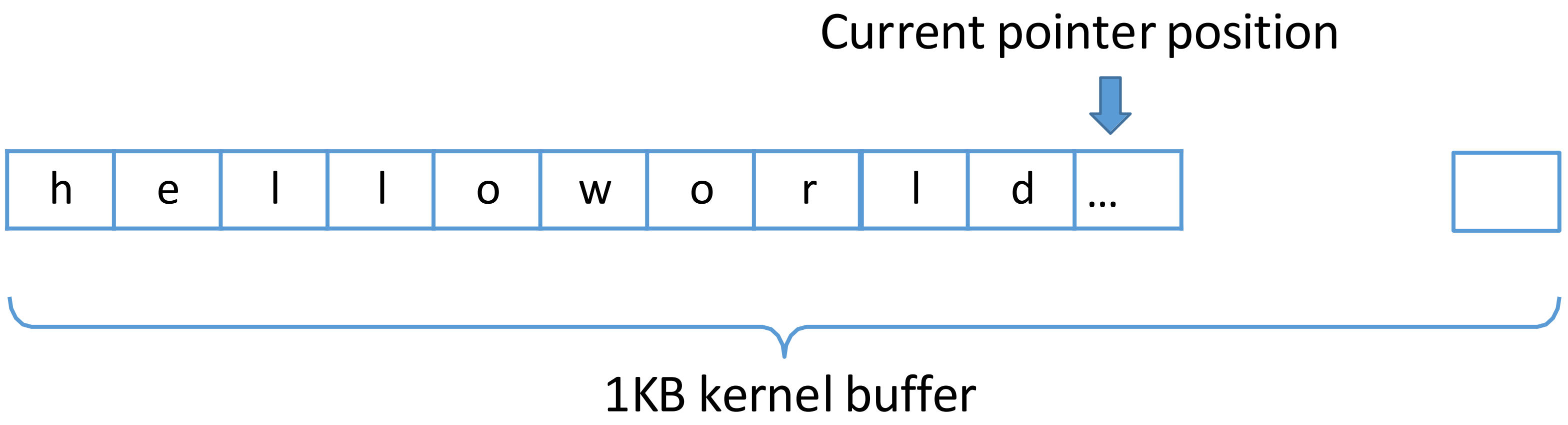
1. Read/write function
    - Input: device file, user-space buffer, offset
    - Output (recommended): the number of bytes you read or write at the end of each function call
        - If error, return -1

2. Seek function
    - Input: device file, offset, whence (= 0, 1, or 2)
    - Output (recommended)
        - If error, return -1
        - If successful, return 0 or positive value

Activity: Seek operations

PA2 – Seek Options

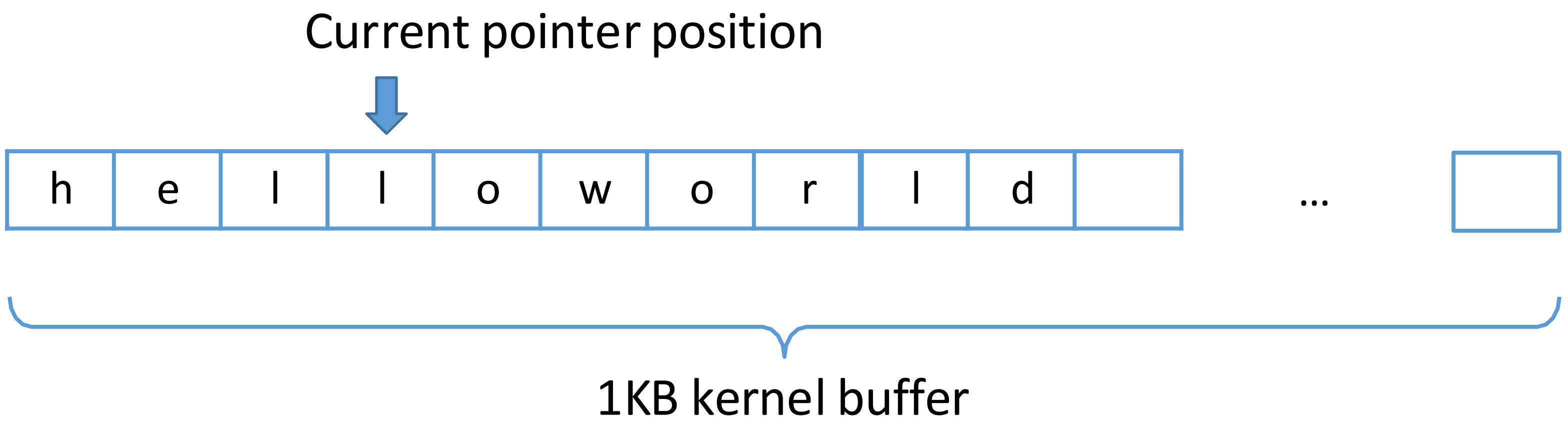
1. SEEK_SET



➔ seek(device_file, 3, 0)

PA2 – Seek Options

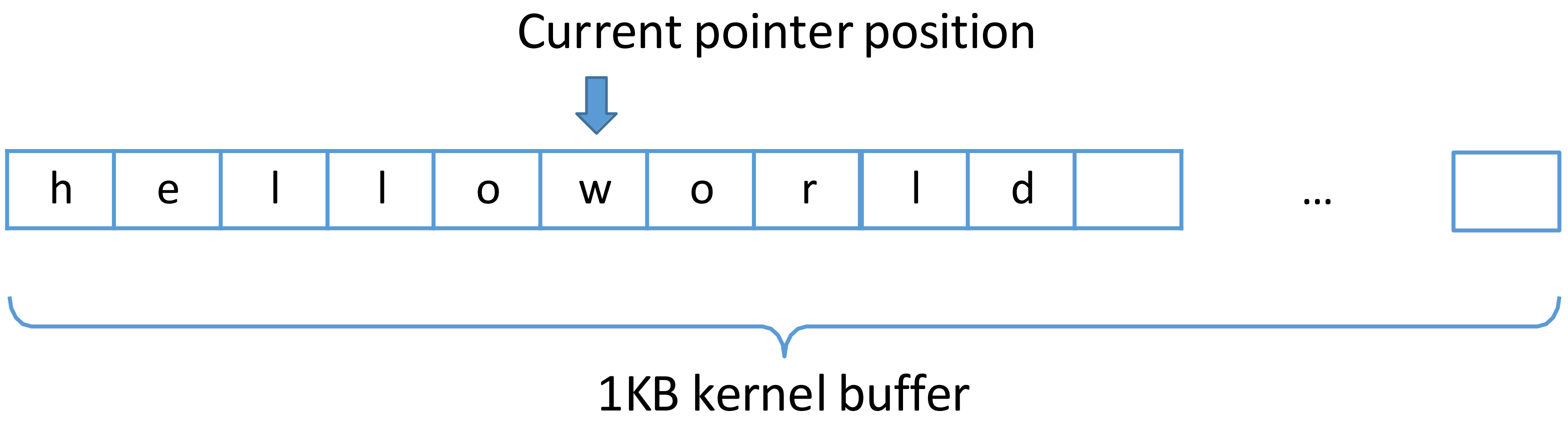
1. SEEK_SET



➔ seek(device_file, 3, 0)

PA2 – Seek Options

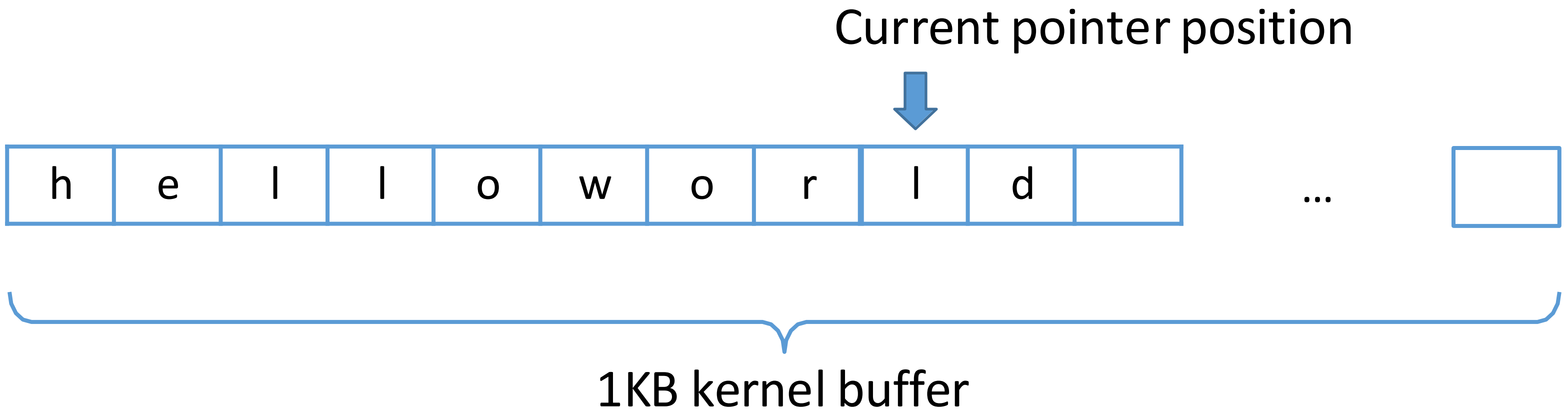
2. SEEK_CUR



➔ seek(device_file, 3, 1)

PA2 – Seek Options

2. SEEK_CUR

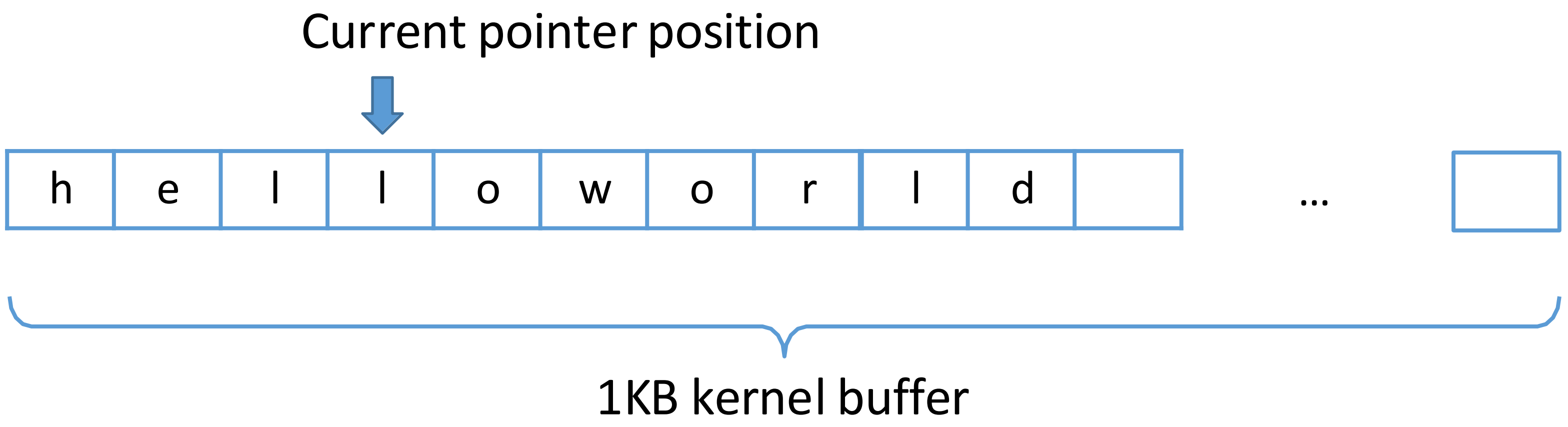


➔ seek(device_file, 3, 1)

➔ seek(device_file, -5, 1)

PA2 – Seek Options

2. SEEK_CUR

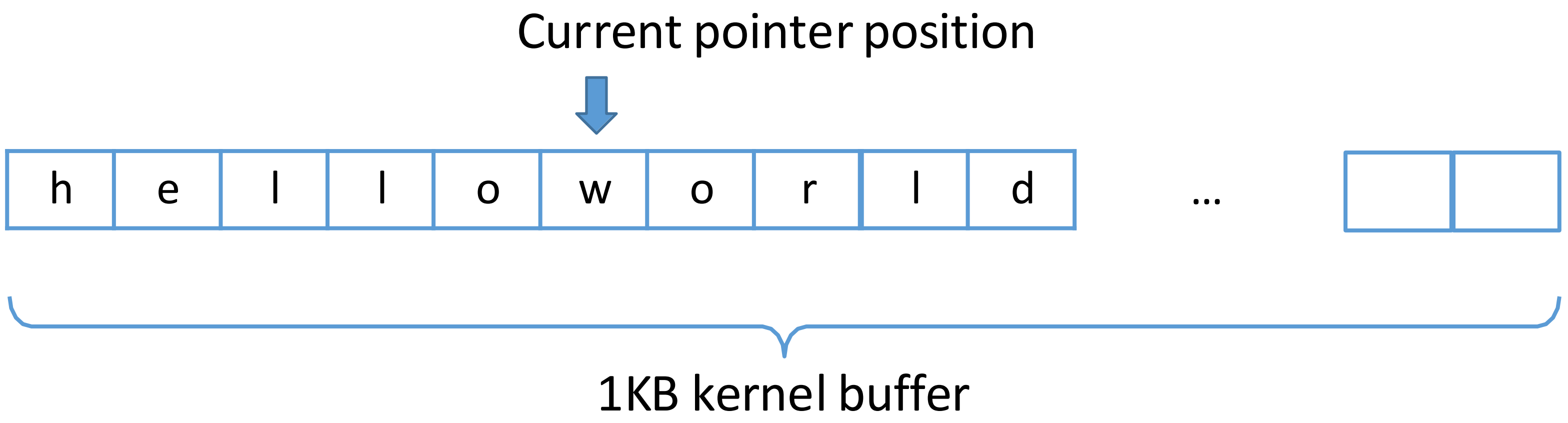


➔ seek(device_file, 3, 1)

➔ seek(device_file, -5, 1)

PA2 – Seek Options

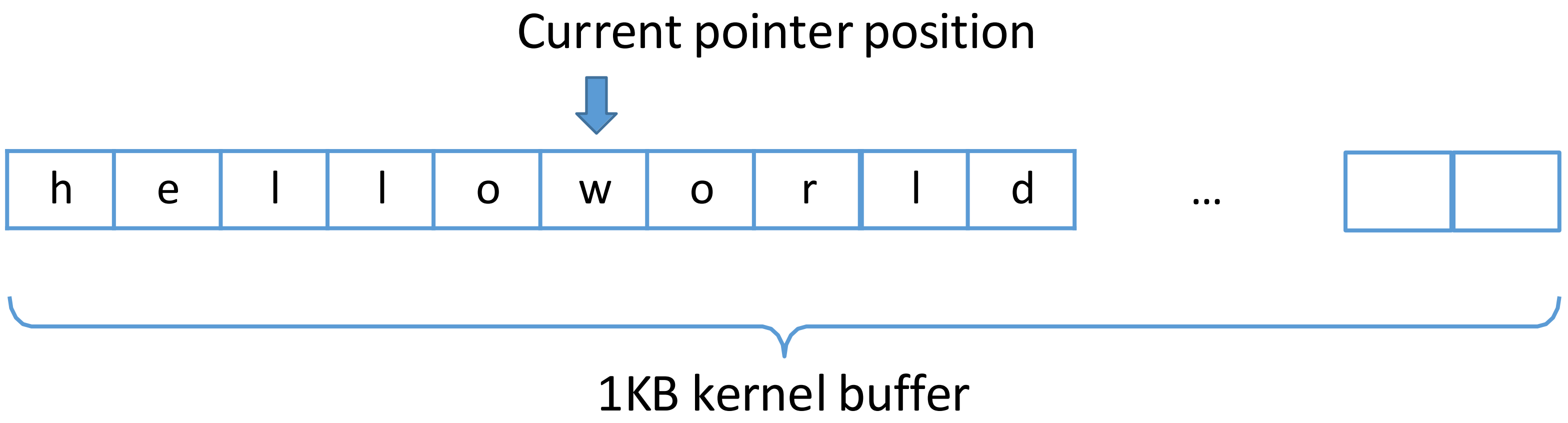
3. SEEK_END



➔ seek(device_file, 1, 2)

PA2 – Seek Options

3. SEEK_END

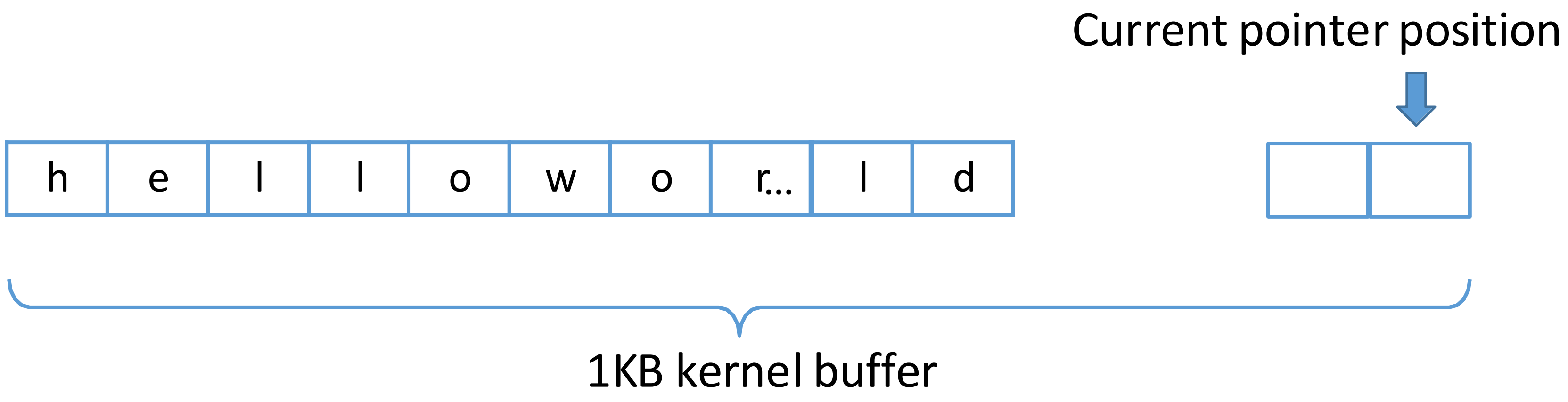


➔ seek(device_file, 1, 2)

➔ seek(device_file, -1, 2)

PA2 – Seek Options

3. SEEK_END



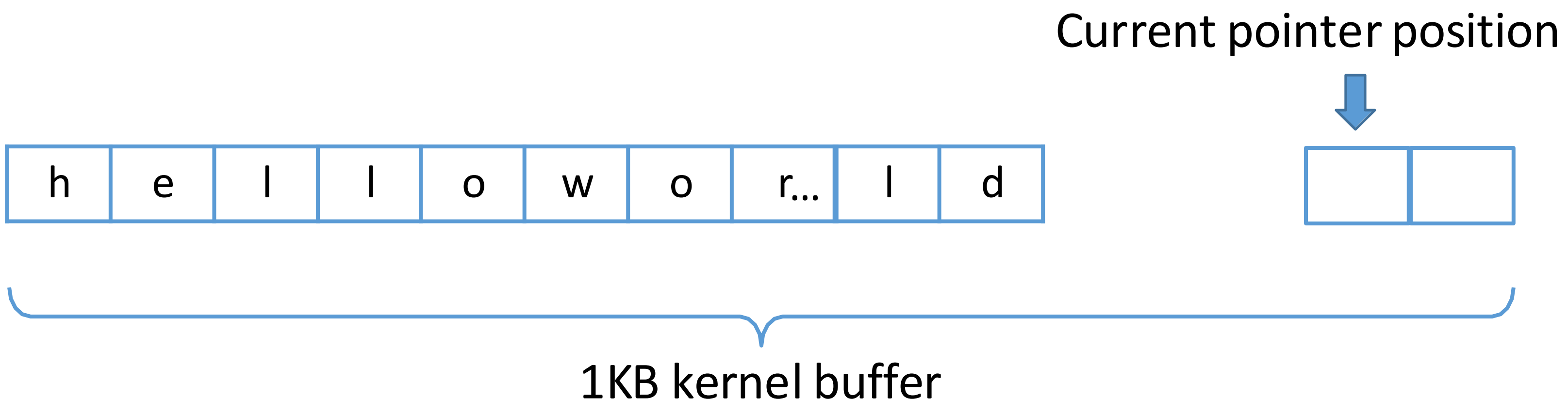
➔ seek(device_file, 1, 2)

➔ seek(device_file, -1, 2)

➔ seek(device_file, -2, 2)

PA2 – Seek Options

3. SEEK_END



➔ seek(device_file, 1, 2)

➔ seek(device_file, -1, 2)

➔ seek(device_file, -2, 2)

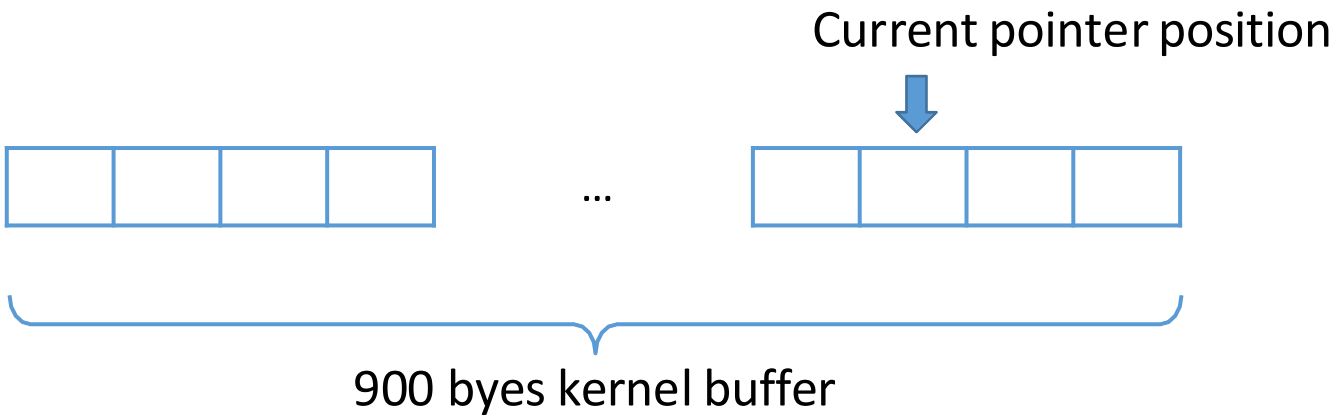
PA3 – Requirements

Dynamically allocate constant-size 900 bytes kernel buffer to store the data written by the user

- kmalloc()
  - Allocate memory for objects smaller than page size in the kernel at initialization time
- kfree()
  - Free memory previously allocated using kmalloc() before exiting

PA3 – Requirements

2. NO over seeking/reading/writing in buffer

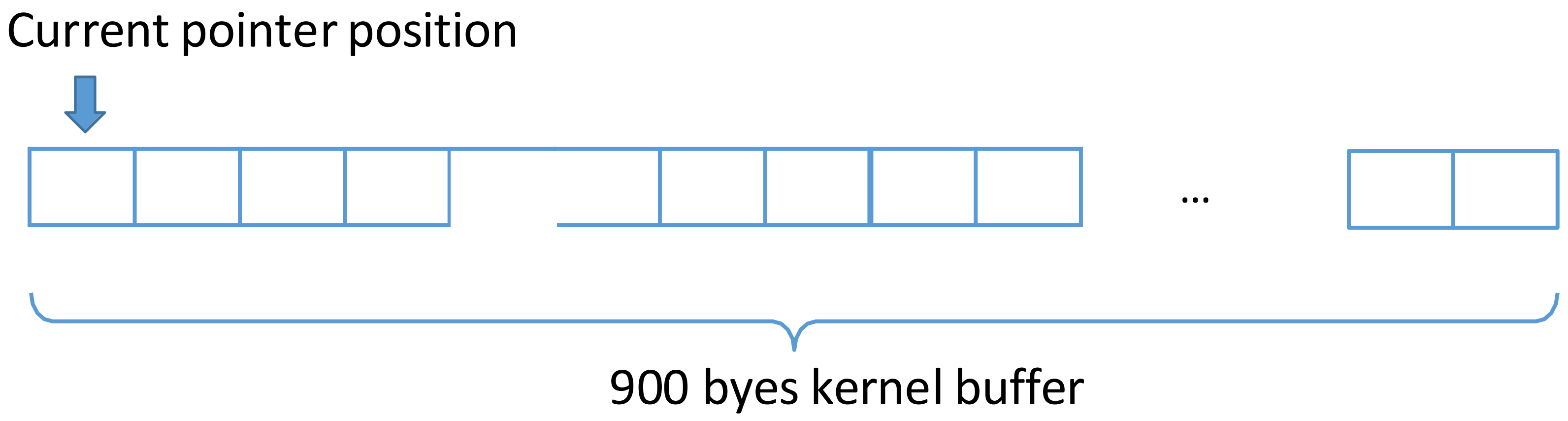


➔ Write("hello")?

➔ Return an error with -1 value and leave the current position unchanged

PA3 – Requirements

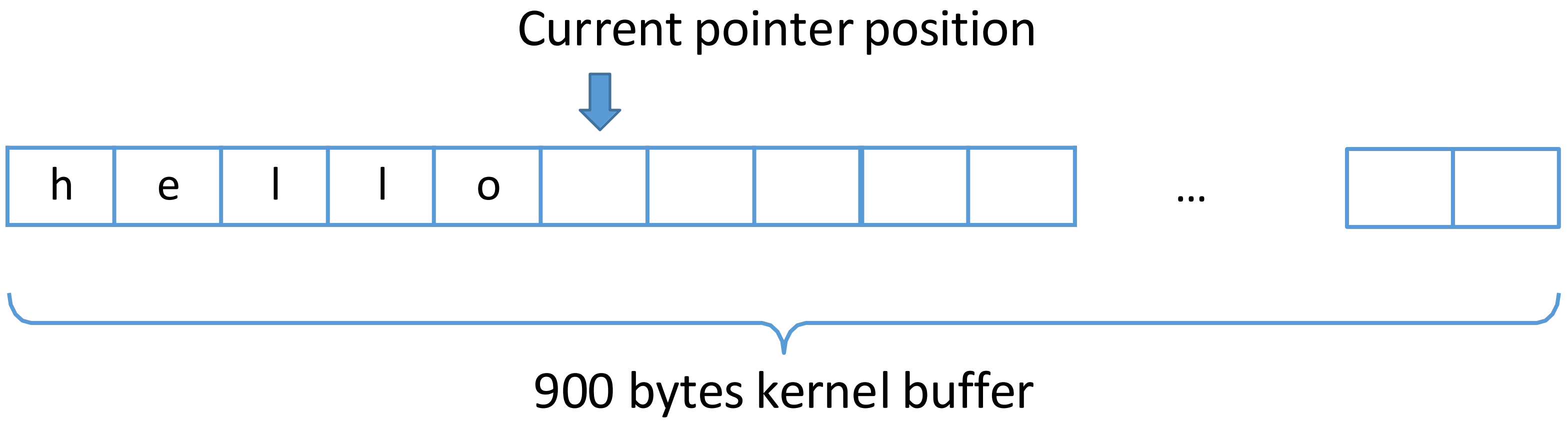
3. Always remember the position of pointer in the device file after each input action



➔ Write("hello")

PA3 – Requirements

3. Always remember the position of pointer in the device file after each input action

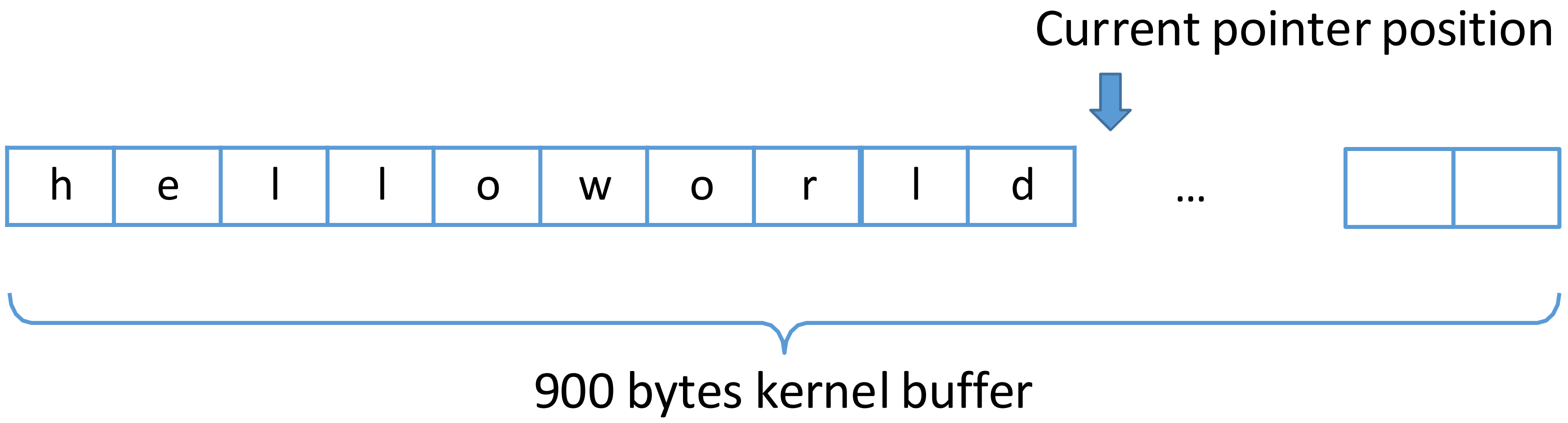


➔ Write(“hello”)

➔ Write(“world”)

PA3 – Requirements

3. Always remember the position of pointer in the device file after each input action



➔ Write("hello")

➔ Write("world")