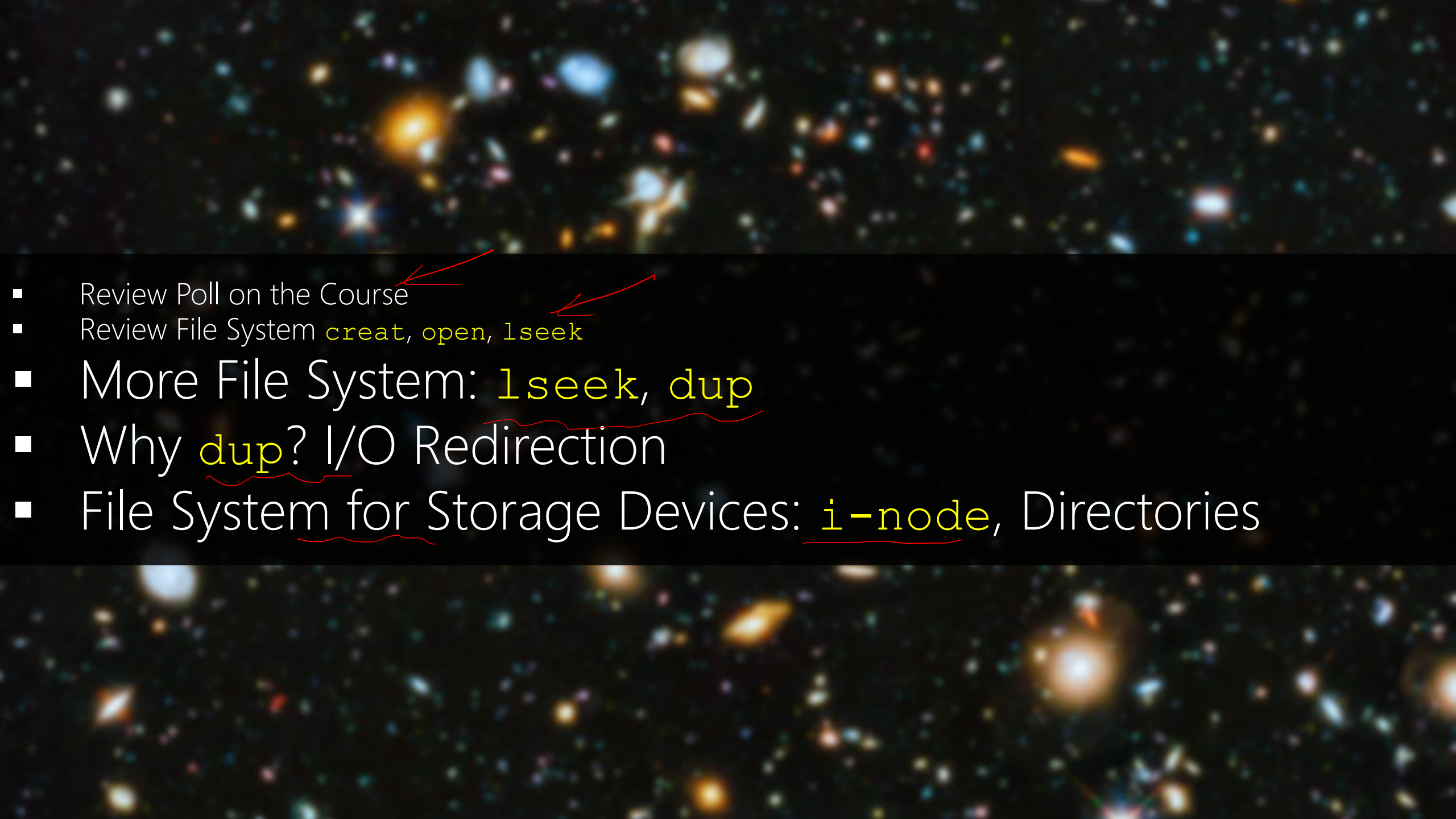
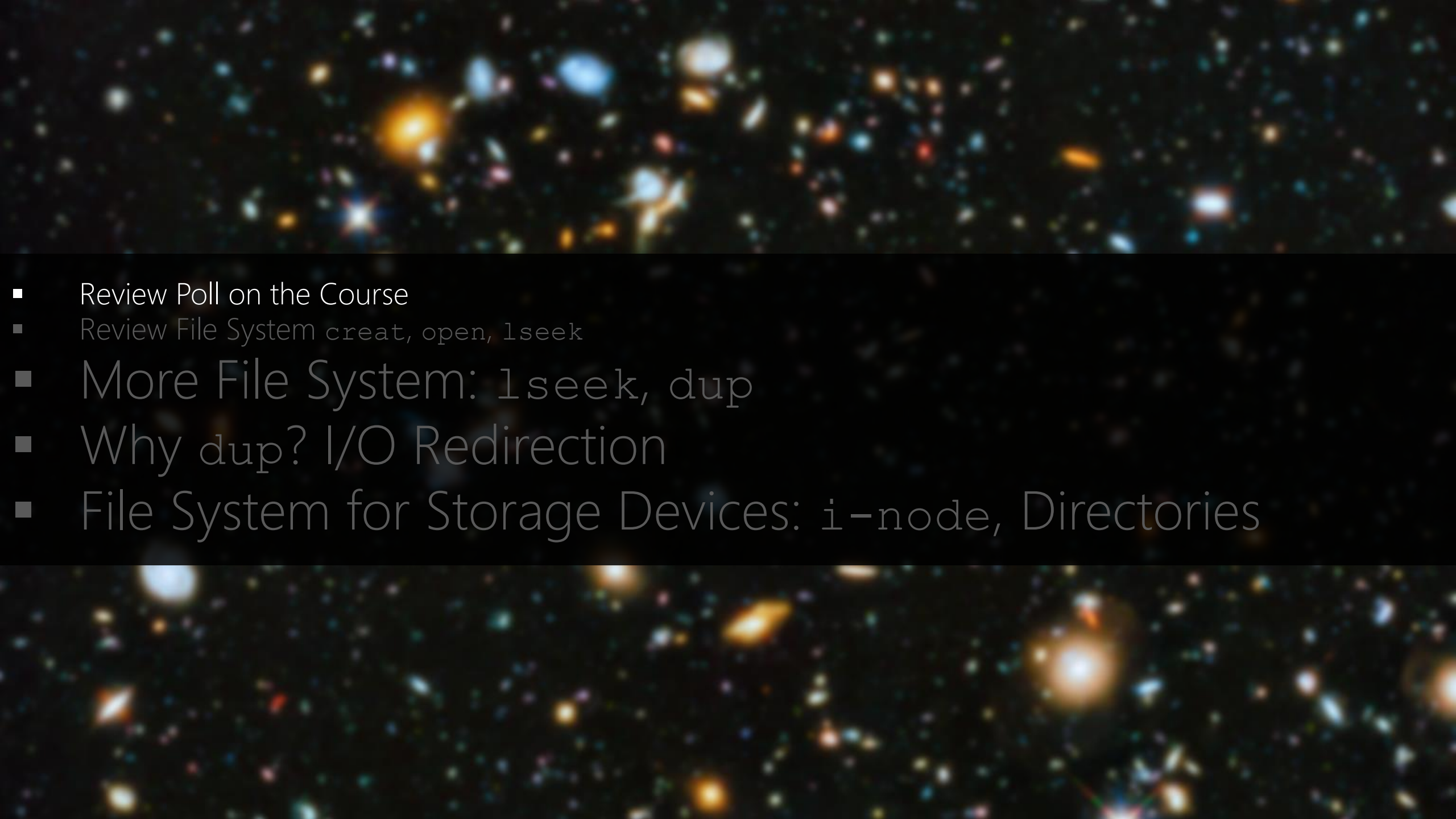


A deep-field astronomical image showing a vast field of galaxies against a black background. The galaxies are of various colors, including yellow, orange, blue, and red, and are scattered across the frame. A horizontal blue line is drawn across the middle of the image, passing behind the text.

Lab06 + Lec06 is up at bb!

- 
- Review Poll on the Course
  - Review File System `creat`, `open`, `lseek`
  - More File System: `lseek`, `dup`
  - Why `dup`? I/O Redirection
  - File System for Storage Devices: `i-node`, Directories



- 
- Review Poll on the Course
  - Review File System `creat`, `open`, `lseek`
  - More File System: `lseek`, `dup`
  - Why `dup`? I/O Redirection
  - File System for Storage Devices: `i-node`, Directories

A deep-field astronomical image showing a vast field of galaxies in various colors (blue, orange, white) against a black background. Two horizontal blue lines frame the central text.

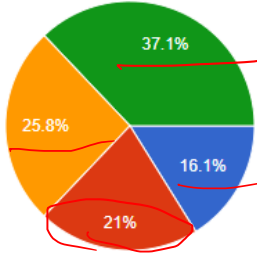
# Poll on the Course

<https://forms.gle/jJzEifEqU68TpZ3z9>

## 62 responses

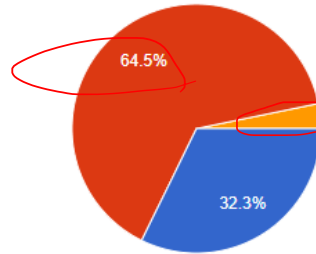
How was the exam?

- Expected more difficult exam
- Expected easier exam
- Easy but long, needed more time
- No complains



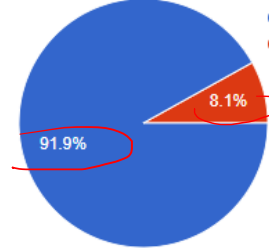
How is it going with the course so far?

- Awesome
- So so
- Bad

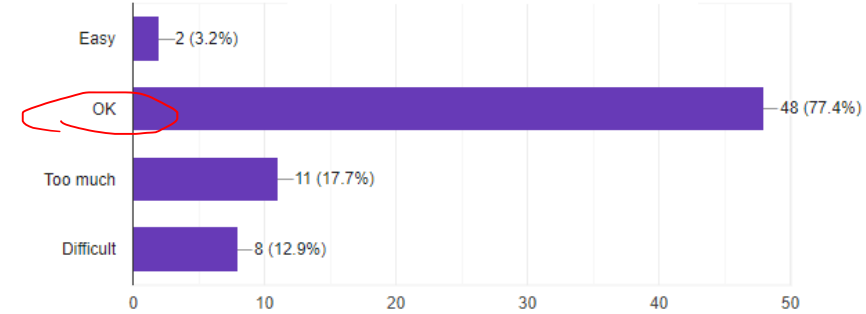


Is it worth taking this course?

- Yes
- No

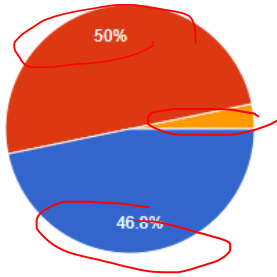


How's the load with the coursework?



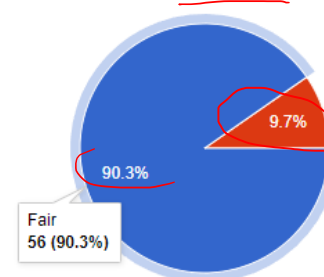
How are GAs/TAs?

- Fully satisfied
- Satisfied
- Not satisfied



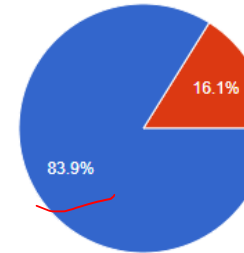
How's marking?

- Fair
- Not fair

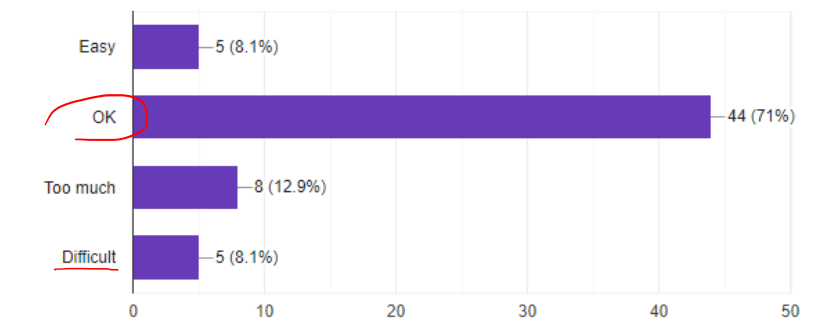


Is it worth doing all this coursework (LABs + LECs):

- Yes
- No

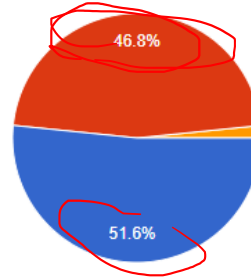


How are LECs (question assignments)?

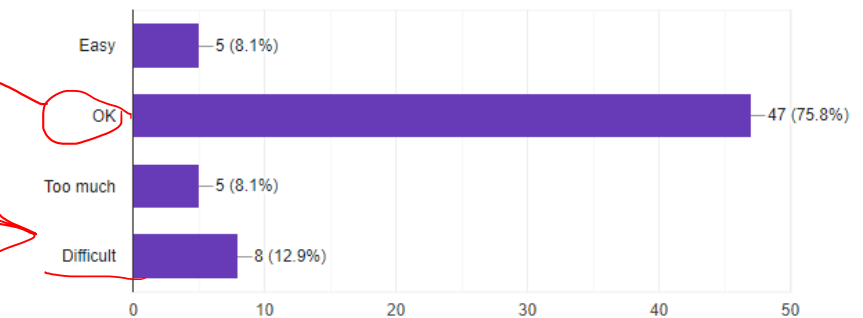


How's Hossein's teaching?

- Awesome, no room for improvement!
- Fair, needs to be improved.
- Awful, cannot be fixed!



How are LABs (programming assignments)?



	3	3	4	4	4	4	Talk slower please.
	3	3	4, 3	4	4	4	Miscellie
	3	3	3	4	4	4	Please mark question on exam that has to do with the music you're recommending????
1	3	2	0	0	0	0	One of the worksheets had like my handwriting :)
3, 4	2	0	0	0	0	0	Slides have too little information and are hard to read off of due to bad background choices. Too many music references.
	3	3	3	4	4	4	Simplify your explanations a little more. This is a really complex course with a deep learning curve.
4, 8	3	3	3	4	4	4	more time on midterms or fewer questions
3, 8	3	3	3	4	4	4	Lecture slides have little writing on them that can be referenced for studying
3, 8	3	3	3	4	4	3	I find the slides difficult to follow after the original lectures. A little bit of context and information on the slides would improve my understanding more.
	3	3	3	4	4	3	more coding demos!
	3	3	3	4	4	3	More time for exam. There are a lot of information that I didn't get to put on my exam. It was fair exam. We had all the information provided through lecture slides and teaching to get ready for exam.
	3	3	3	4	4	3	Sometimes the perspective/slides about formal is hard to follow
4	2	3	4	0	0	3	exam questions were to spread ended and required us to come up with solutions that were not covered in class.
	3	3	3	4	4	4	Assume names and initials
	3	3	3	4	4	4	In class, I feel like sometimes things are mentioned as common knowledge but they are not so I have to research lectures and search things up to make sure I understand. Of course, not every detail needs to be explained but when something new is introduced, it would be nice to explain it
	3	3	3	4	4	4	Although the midterm wasn't hard, studying for it was hard because I did not know what to expect. Identifying the different concepts required a bit more explicitly during lectures could be helpful, especially since the slides cannot be used to identify all those topics. Enjoying the dialog
	3	3	3	4	4	3	I enjoyed the unique teaching style that you offer Dr. Fanti, however it is still very helpful to have a bit more information on the slides, even if everything you say is accurate! But it's sometimes hard to look over the slides alongside my own notes (especially with stuff you write in class
	3	3	3	4	4	3	The slides don't have much context to them, so it's hard to know what each lecture slide is about.
4	2	3	4	0	0	3	The exam questions were to spread ended which makes it difficult coming up with a solution when the topics are not covered in lectures as in the textbook. Also, slides could include more context.
4	2	3, 8	0	0	0	3	More exam preparation would be helpful, and also please talk slower.
4	2	3	0	0	0	3	There should be smaller amount of lecture slides but with more information. I go for studying through the slides but too many times they have only one word. I would prefer slides with a bit more information. Also, the length of the lectures could be reduced as it is harder to focus on
	0	0	0	0	0	0	more details in slides
	3	3	3	4	4	3	I don't think this is necessary for midterms. But for the exam, I think we should have practice exams because it will be a lot more difficult than the midterms and needs a lot more studying.
	3	3	3	4	4	4	More clear wording on the exam
3, 4	2	3	4	0	0	3	Hard to tell unless a lot of research was done confusion.
	3	3	3	4	4	3	I appreciate that the lectures are recorded so that we may watch them at our own schedule. If I had to think of something to make it better, I would say maybe try to repeat a question or answer that is given by a student. It is hard to hear what the students say on the recording, so it would
	3	3	3	4	4	4	good
	3	3	3	4	4	3	Dr. Fanti's style of teaching is different from Computer Science, but fair.
	3	0	3	4	4	3	Lecture slides are a little bit hard to follow.
	3	3	3	4	4	3	Ensuring that lectures end on time and don't run past the scheduled end of the class time
4	0	2	0	0	0	3	The lecture slides have very little info
	0	3	0	4	4	4	Thank you
	3	3	3	4	4	3	Would like some sample question for the exam
	3	0	4	4	4	4	too much importance placed on the dialogue of lectures, a lot of things missed
	0	3	3	0	4	3	Lecture slides are too vague.
	3	3	3	4	4	4	You're a great professor
	3	3	3	4	4	3	Most of the information on the lab assignments such as environment setup aren't needed. They can be useful for students who don't know how to use a computer so I think they should be placed in a separate [optional] pdf and not clutter the actual important information for the assignm
	3	3	3	4	4	3	The process of people bringing more casual-like in only quickboard by <del>random selection</del> form for each class after the midterm for each :) )
	3	3	3	4	4	4	More concise lecture slides
3, 4	3	3	3	4	4	3	Although researching the lectures makes my understanding of the course much clearer, it feels very easy for students to miss the overall scope of the lectures. This is because there are many topic points presented in class, and so it becomes easy for students to get overwhelmed. In
	3	3	3	4	4	4	Make the lecture slides be more organized. Some of them just look like duplication and don't have much information on them.
	3	0	0	4	4	4	Lectures are extremely difficult to follow
	3	3	3	4	4	4	want to do more code work ^_<
4	4	4	4	4	4	3	Better music taste
4	4	2, 8	0	0	0	4	Some more study guides for the exam would have been nice
	3	3	3	4	4	4	Keep doing good!
	3	3	3	4	4	3	Some of the lab assignment questions need to be written more clearly
	3	3	3	4	4	4	This isn't really a criticism but the slides in the lecture could use some more text. Some teachers have slides which are too text heavy, but this course is too text light sometimes. It makes revision difficult because you don't have the context of what the slides mean, if you want to understand
	3	0	0	4	4	4	
	3	3	3	4	4	4	Giving the class online attendance options via k should continue and if other parts like exams can be done [if school allows] then that should move forward too. Lecture is good but slides should have at least little more info as it would help with following along.
	0	3	0	4	4	4	Not teaching Dash script made me very confused and slow for long time on lab4
	3	2	3	4	4	4	Walking is specific
	3	3	3	4	4	3	I didn't find some of the exam questions clear enough for specifying exactly what you wanted as input as the answers. My answers for some of the opinionated questions such as questions 6 and 9 were completely different than what you put as your exam key.
	3	4	3	4	4	4	Would have liked a sample test for the midterm.
	3	3	2	4	4	4	Please add more writing to your slides, even if you don't want to read it.
	3	3	2	0	0	3	Really thank shell scripting before assigning it.
4	4	4	4	4	4	4	I'm just scared because you said it's going to get a lot harder after reading work
	3	3	3	4	4	3	

The process of people becoming more robot-like is only quickened by a reused review form for each class after the midterm for each. :-)

Better movie taste

I'm just scared because you said it's going to get a lot harder after reading week 🤔🤔🤔

I understand the stance on not having more content in the slides but for some people having point form information helps greatly to understand what you're teaching. It's hard to follow with just titles and pictures from movies. I love the effort to have movie analogies to relate the class topic to the students but I find that most times it just confuses me more. I find most classes don't give me as much information as going home and googling it myself and while being able to find this information on your own is important, I come to class hoping to learn from there as it should be more reliable than the internet resources. Not to mention google won't teach me the information that you may find important on an exam. So then I'm learning things that might not even apply to the course or be useful for me at all. I find the lectures with point form, or summarized information of the topic are a very useful study guide giving me important information I should understand, helps me to follow what's being discussed in the lecture visually, and to guide my studying later on for exams and such. I know for example in COMP-2660 the professor lines her labs up pretty nicely with the lectures so it's almost as if it's review and it makes it so much easier to study and memorize what I need for the quizzes and tests. I found that the information sticks much better using this method. The labs also include a questions portion as well as an applied coding portion so it gives the same idea as a lec and lab assignment, not too much, and it all helps to further understand and review the topics.



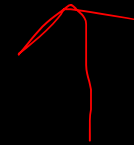
# Lectures

- Slides
  - More written notes, less slides  
→ Already told/warn you! Will try but won't replace lectures!
  - More organization, hard to follow  
→ Sure! I need suggestions
- Class
  - Talk slower, needs rewatch  
→ True. Donno how to fix it. Many topics, limited timing ...
  - Finish on time → Agree. I'm trying ...
  - More coding → No! we have labs.



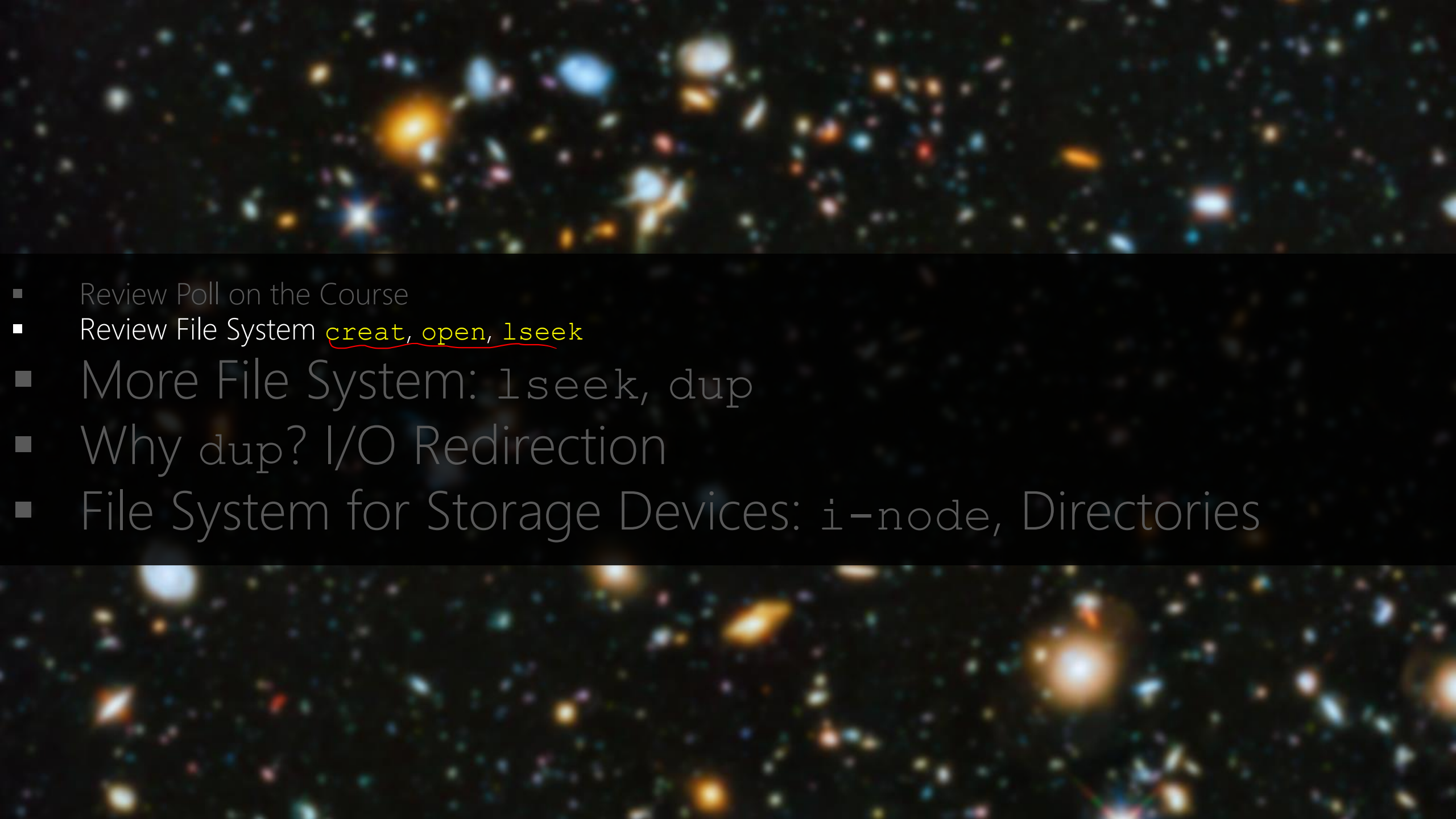
## Assignments

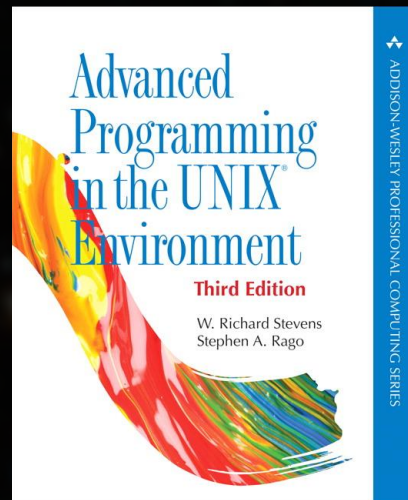
- Better wording → Sure! I need suggestions
- Not covered in class (e.g., shell scripting) → Purposefully, we designed it like this.
- Some unnecessary trivial content → It's subjective. We had comments for more details in the past



## Exam

- Not covered in class → Like what?
- Sample exams → Lecture assignments.
- Better wording → Agree, but I need suggestions.
- May have wide range of answers → Actually, we want this.  
Sometimes we offer bonus points for novel or a different answer.  
Talk to your lab instructor if you lost marks.

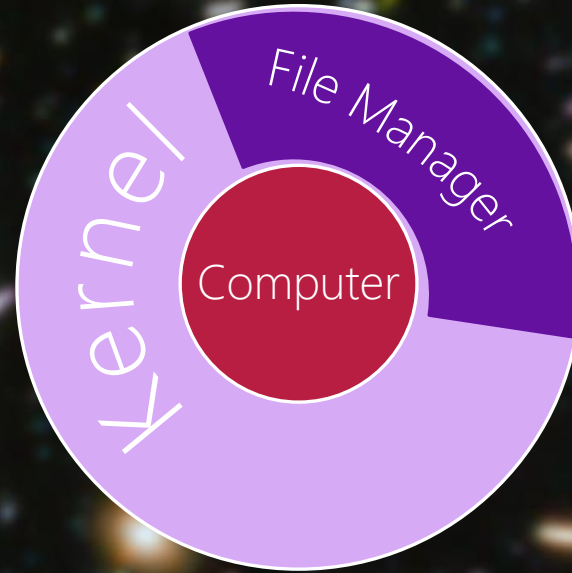
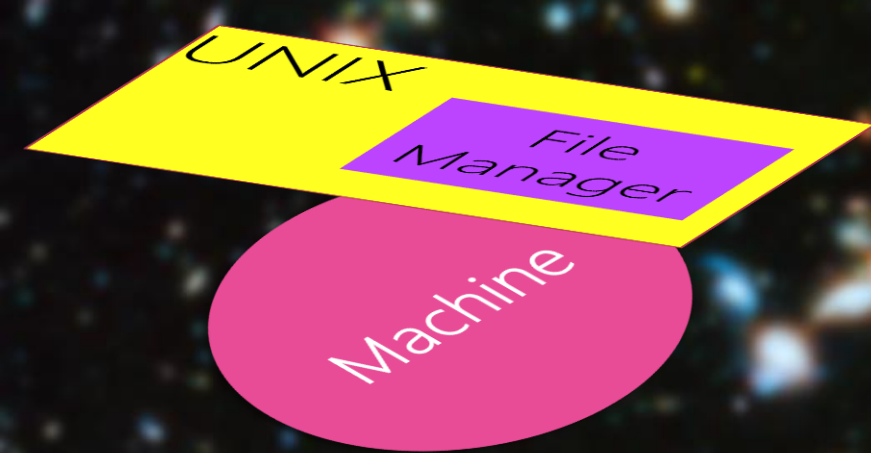
- 
- Review Poll on the Course
  - Review File System creat, open, lseek
  - More File System: lseek, dup
  - Why dup? I/O Redirection
  - File System for Storage Devices: i-node, Directories



} Chapter 03: File I/O }

} Chapter 04: File I/O }





# Computer

Memory

Kernel: Device Manager

Kernel: Memory Manager

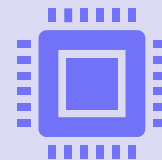
Kernel: File Manager

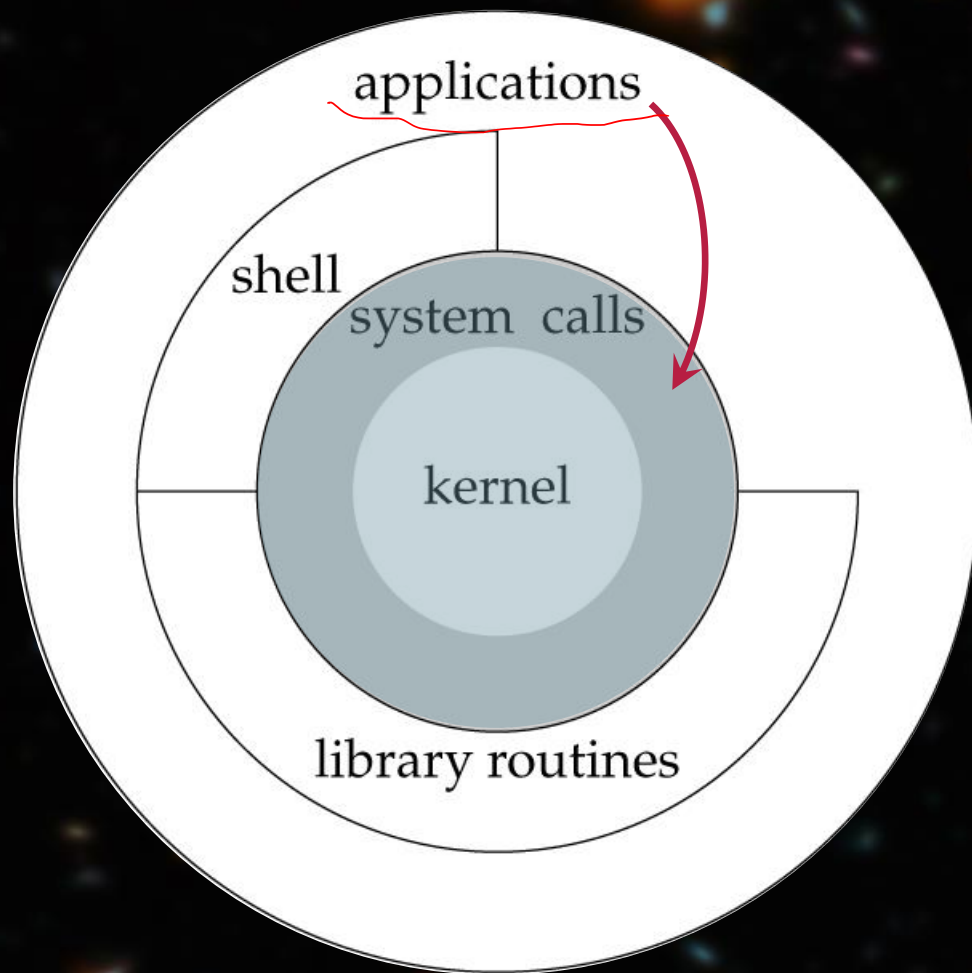
Kernel: Network Manager

Kernel: Process Manager

Bus

Processor





Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
< aio.h>	•	•	•	•	asynchronous I/O
< cpio.h>	•	•	•	•	cpio archive values
< dirent.h>	•	•	•	•	directory entries (Section 4.22)
< dlfcn.h>	•	•	•	•	dynamic linking
< fcntl.h>	•	•	•	•	file control (Section 3.14)
< fnmatch.h>	•	•	•	•	filename-matching types
< glob.h>	•	•	•	•	pathname pattern-matching and generation
< grp.h>	•	•	•	•	group file (Section 6.4)
< iconv.h>	•	•	•	•	codeset conversion utility
< langinfo.h>	•	•	•	•	language information constants
< monetary.h>	•	•	•	•	monetary types and functions
< netdb.h>	•	•	•	•	network database operations
< nl_types.h>	•	•	•	•	message catalogs
< poll.h>	•	•	•	•	poll function (Section 14.4.2)
< pthread.h>	•	•	•	•	threads (Chapters 11 and 12)
< pwd.h>	•	•	•	•	password file (Section 6.2)
< regex.h>	•	•	•	•	regular expressions
< sched.h>	•	•	•	•	execution scheduling
< semaphore.h>	•	•	•	•	semaphores
< strings.h>	•	•	•	•	string operations
< tar.h>	•	•	•	•	tar archive values
< termios.h>	•	•	•	•	terminal I/O (Chapter 18)
< unistd.h>	•	•	•	•	symbolic constants
< wordexp.h>	•	•	•	•	word-expansion definitions
< arpa/inet.h>	•	•	•	•	Internet definitions (Chapter 16)
< net/if.h>	•	•	•	•	socket local interfaces (Chapter 16)
< netinet/in.h>	•	•	•	•	Internet address family (Section 16.3)
< netinet/tcp.h>	•	•	•	•	Transmission Control Protocol definitions
< sys/mman.h>	•	•	•	•	memory management declarations
< sys/select.h>	•	•	•	•	select function (Section 14.4.1)
< sys/socket.h>	•	•	•	•	sockets interface (Chapter 16)
< sys/stat.h>	•	•	•	•	file status (Chapter 4)
< sys/statvfs.h>	•	•	•	•	file system information
< sys/times.h>	•	•	•	•	process times (Section 8.17)
< sys/types.h>	•	•	•	•	primitive system data types (Section 2.8)
< sys/un.h>	•	•	•	•	UNIX domain socket definitions (Section 17.2)
< sys/utsname.h>	•	•	•	•	system name (Section 6.9)
< sys/wait.h>	•	•	•	•	process control (Section 8.6)

```
fd = creat()
```



```
graph TD; A["fd = creat()"] --> B["write(fd, sth)"]; B --> C["close(fd)"]
```

A vertical flowchart with three green rectangular boxes. The top box contains the code `fd = creat()` with `fd` and `creat` underlined in red. A yellow arrow points down to the middle box, which contains `write(fd, sth)` with `write` underlined in red. Another yellow arrow points down to the bottom box, which contains `close(fd)` with `close` underlined in red.

```
write(fd, sth)
```

```
close(fd)
```



```
graph TD; A["fd = open(creat if not exists)"] --> B["read(fd, sth)"]; A --> C["write(fd, sth)"]; B --> D["close(fd)"]; C --> D
```

A flowchart starting with a green box at the top containing `fd = open(creat if not exists)`, where `fd` and `creat` are underlined in red. A red arrow points down to this box from above. Two yellow arrows branch out from the box to two blue boxes below: `read(fd, sth)` on the left and `write(fd, sth)` on the right, both with `read` and `write` underlined in red. Two yellow arrows then converge from these blue boxes to a final green box at the bottom containing `close(fd)`.

fd = open(creat if not exists)

```
read(fd, sth)
```

```
write(fd, sth)
```

```
close(fd)
```



Only one of them

<u>O_RDONLY</u>	Open for reading only (the returned fd can only read)
<u>O_WRONLY</u>	Open for write only (the returned fd can only write like <code>creat()</code> )
<u>O_RDWR</u>	Open for reading and writing (the returned fd can do both read and write)
<u>O_EXEC</u>	Open for execute only (the returned fd execute)
<u>O_SEARCH</u>	Open for search only (for directories)

```
#include <fcntl.h>
int open(const char *path, int oflag, ...);
```

non-negative number (fd) if OK  
-1 on error



## In combination with other flags

---

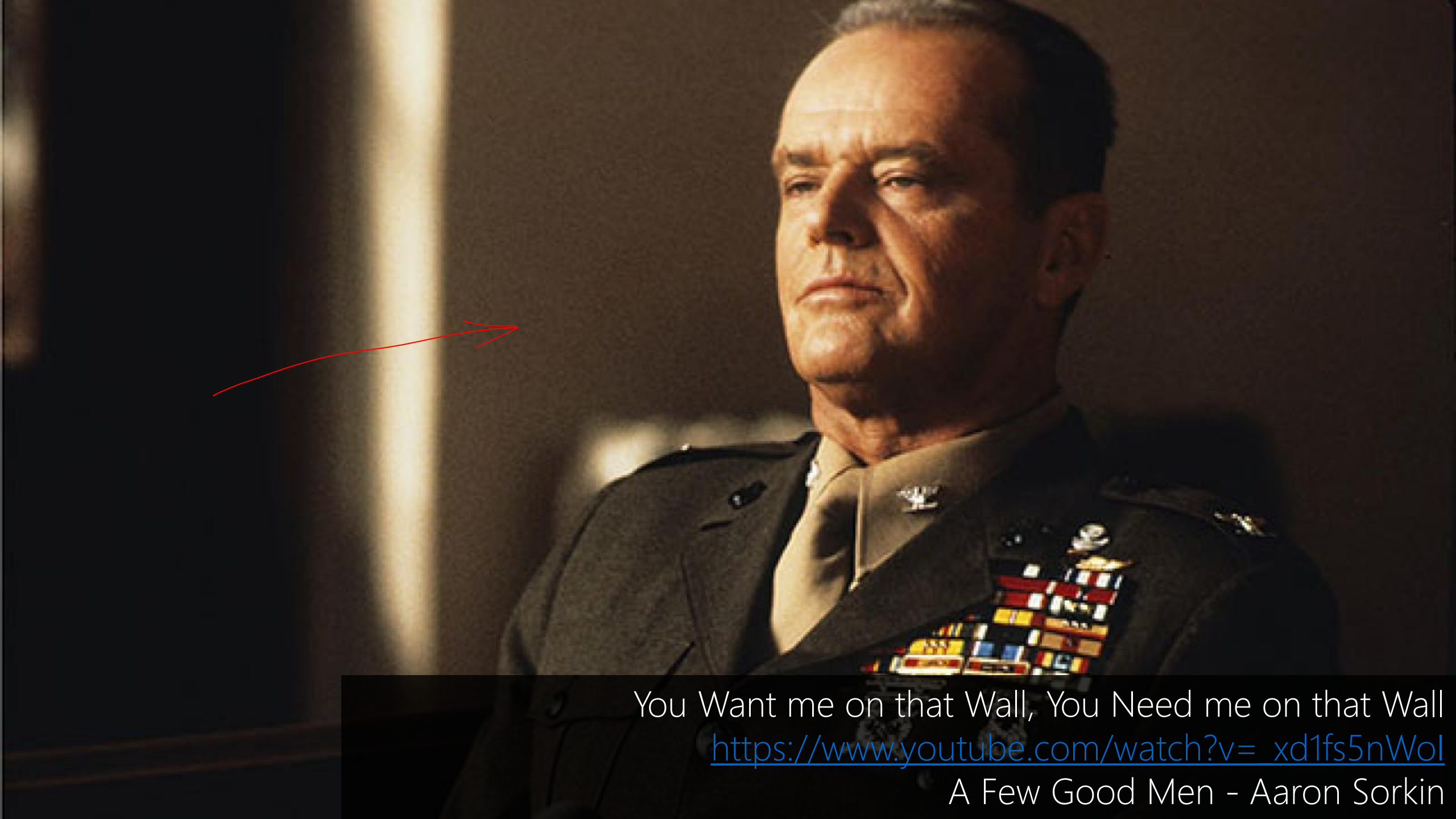
O_CREAT	}	Create the file if it does not exist (you have to specify the mode_t)
O_EXCL		Raise error (fd == -1) if create and the file already exists

<https://stackoverflow.com/questions/48388212/what-is-the-written-out-word-for-o-excl>

---

```
#include <fcntl.h>
int open(const char *path, int oflag, ...);
```

non-negative number (fd) if OK  
-1 on error



You Want me on that Wall, You Need me on that Wall

<https://www.youtube.com/watch?v=xd1fs5nWol>

A Few Good Men - Aaron Sorkin

## lseek

POSIX

```
#include <unistd.h>
int lseek(int fd, off_t offset, int whence);
```

file's new offset if OK (can be negative)  
-1 on error



---

Every opened file has one sentinel: `current offset`  
Measures the number of bytes from the beginning of the file.

---

`creat()` or `open()` → 0



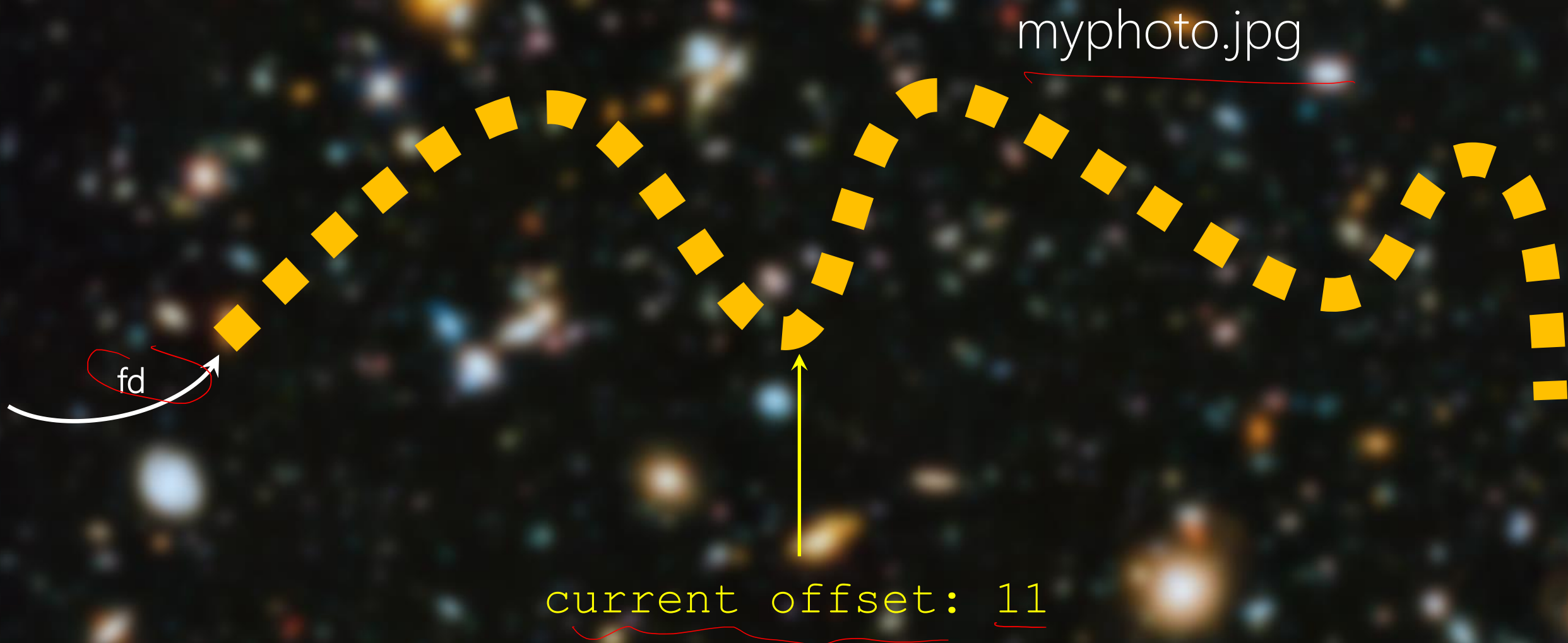
---

Every opened file has one sentinel: `current offset`  
Measures the number of bytes from the beginning of the file.

---

`read()` or `write()` → ++ actual number of bytes read or written

`read()` or `write()` → always move forward



---

## lseek

How many bytes move the current offset

---

```
#include <unistd.h>
int lseek(int fd, off_t offset, int whence);
file's new offset if OK (can be negative)
-1 on error
```

```
#include <sys/types.h>
typedef off_t signed long
```

# lseek

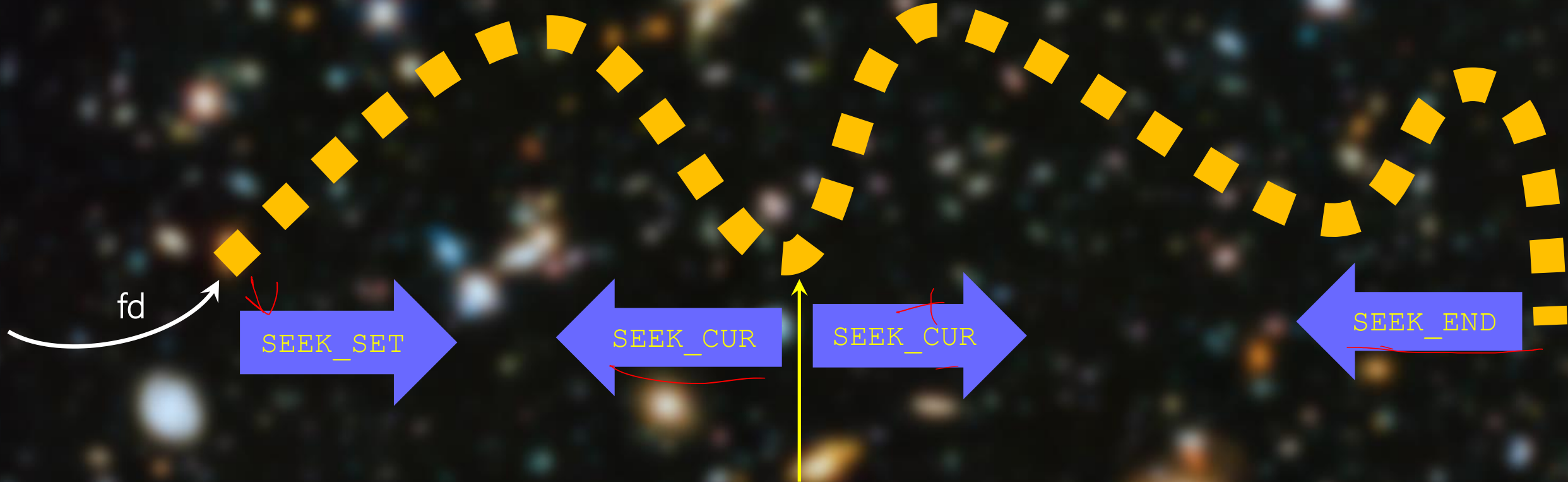
How many bytes move the current offset from what place or origin?

SEEK\_SET, SEEK\_CUR, SEEK\_END

```
#include <unistd.h>
int lseek(int fd, off_t offset, int whence);
file's new offset if OK (can be negative)
-1 on error
```



myphoto.jpg



current offset: 11

## lseek

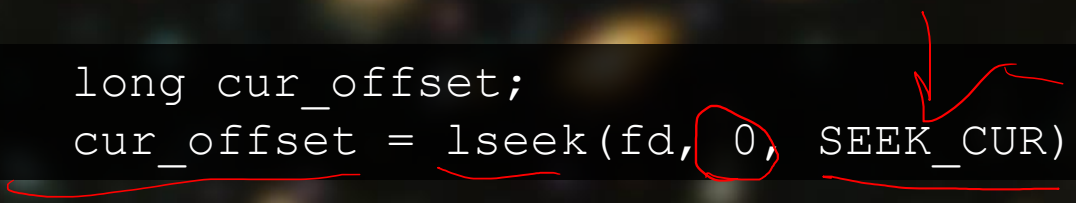
New current offset

```
#include <unistd.h>
int lseek(int fd, off_t offset, int whence);
file's new offset if OK (can be negative)
-1 on error
```

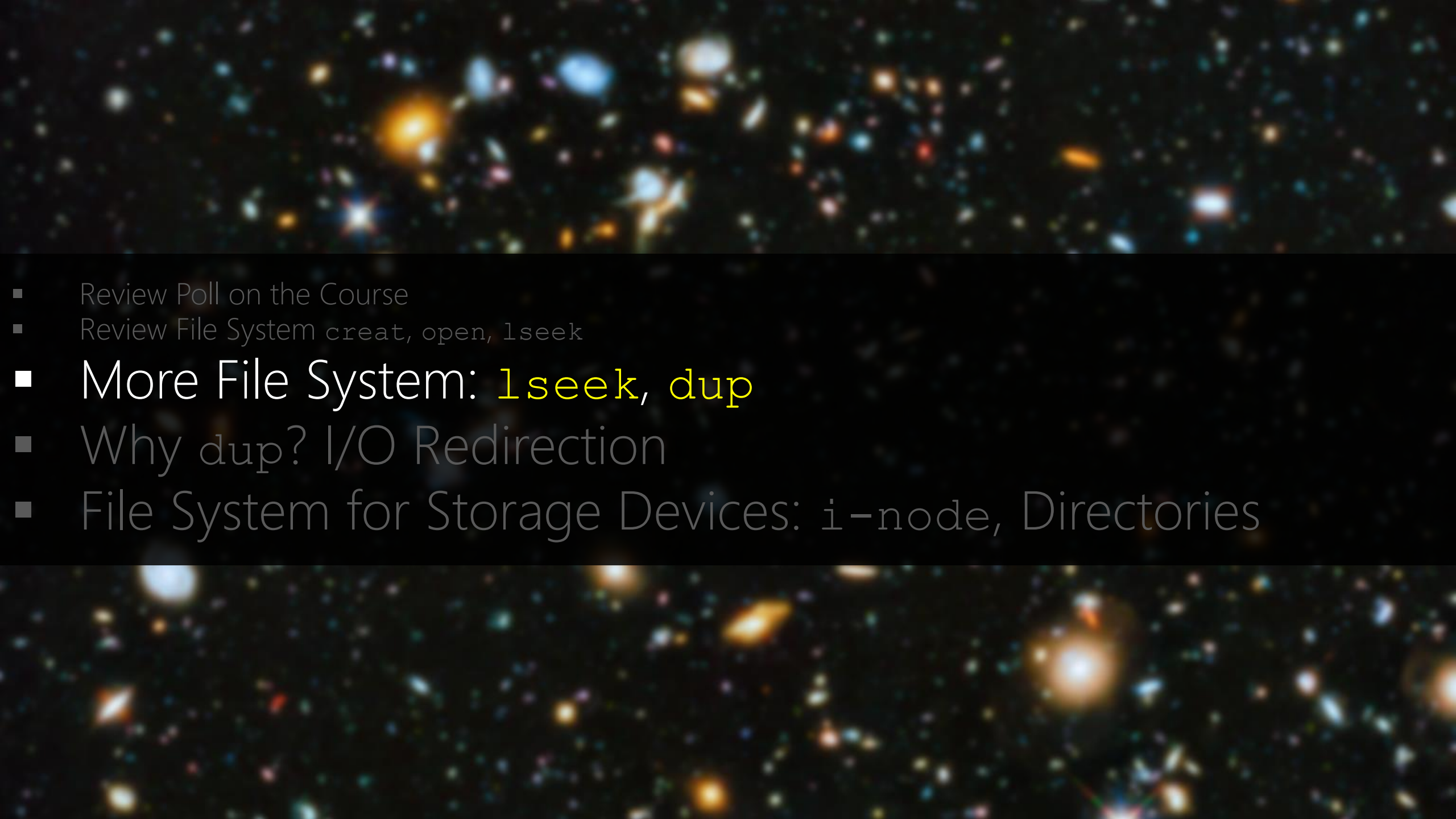
# lseek

How to know the value of current offset?

```
long cur_offset;  
cur_offset = lseek(fd, 0, SEEK_CUR);
```





- 
- Review Poll on the Course
  - Review File System `creat`, `open`, `lseek`
  - More File System: `lseek`, `dup`
  - Why `dup`? I/O Redirection
  - File System for Storage Devices: `i-node`, Directories



# lseek

How to know the file can be seekable?

```
long cur_offset;  
cur_offset = lseek(fd, 0, SEEK_CUR);  
if (cur_offset == -1){  
    /* either the fd is on a file which is not seekable  
       or an error occurred */  
}
```

```
hfani@charlie:~$ tty
/dev/pts/12
hfani@charlie:~$ vi seekable.c
```

```
#include <fcntl.h>
#include <unistd.h>
void main(void)
{
    int fd = open("/dev/pts/12", O_RDWR);
    char error[20] = "cannot seek!\n";
    if (lseek(fd, 0, SEEK_CUR) == -1) {
        write(fd, error, 20);
        return;
    }
}
```

What is tty? Lab06  
Is terminal device seekable?

```
hfani@charlie:~$ cc seekable.c -o seekable
hfani@charlie:~$ ./seekable
cannot seek!
```

Is terminal device seekable? No!

Try other devices, like,

→ /dev/mem  
→ /dev/cpu  
→ /dev/sda/

## 1seek

Is it possible for offset be greater than file size?



# lseek

Is it possible for offset be greater than file size?  
Yes! The next `write()` operation creates a hole of 0s.





```
hfani@charlie:~$ vi hole.c
```

```
#include <fcntl.h>
```

```
#include <unistd.h>
```

```
void main(void){
```

```
    int fd = open("./hole_test.txt", O_RDWR | O_CREAT, S_IRUSR | S_IWUSR);
```

```
    int cur_offset = lseek(fd, 10, SEEK_SET);
```

```
    char buf[20] = "write after the hole.";
```

```
    write(fd, buf, 20);
```

```
}
```

current offset = 0

move it 10 bytes ahead from start

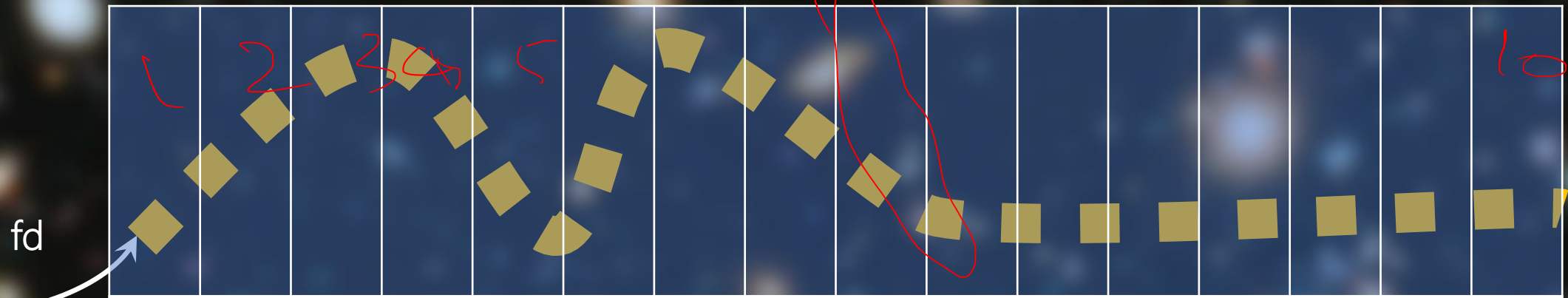
write new bytes



lseek

A Better Use Case: Binary Search in Sorted Elements

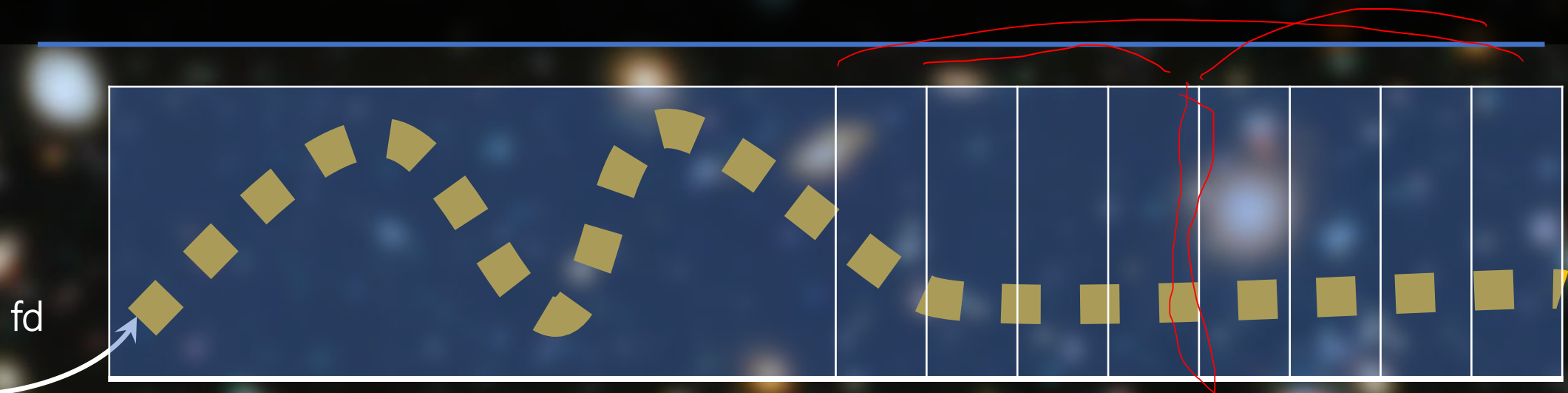
$\mathcal{O}(n) \log$   
 $\mathcal{O}(\log n)$



---

# lseek

A Better Use Case: Binary Search in Sorted Elements

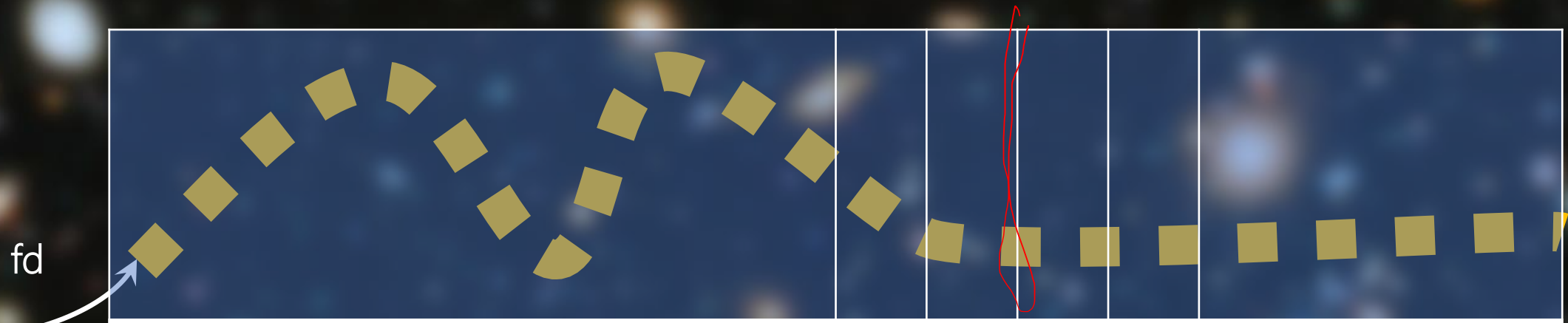




---

# lseek

A Better Use Case: Binary Search in Sorted Elements

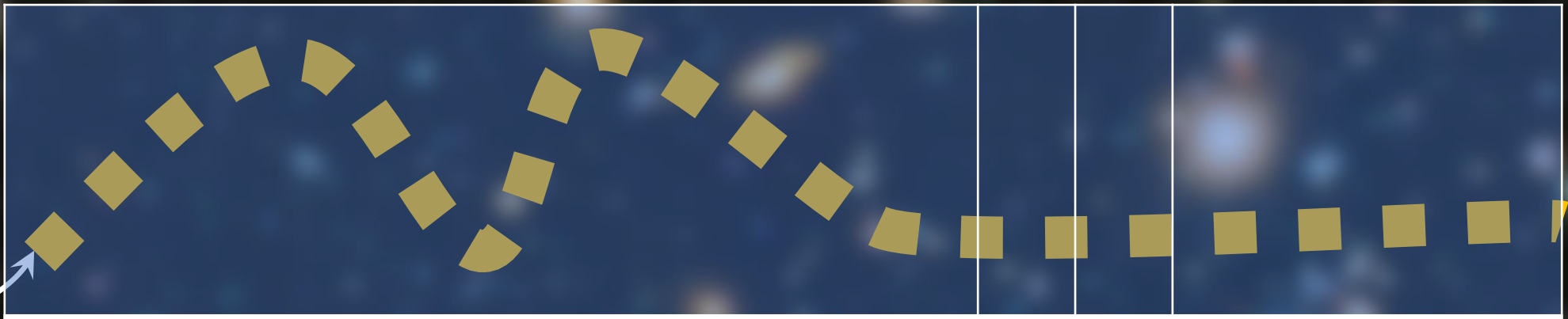


---

# lseek

A Better Use Case: Binary Search in Sorted Elements

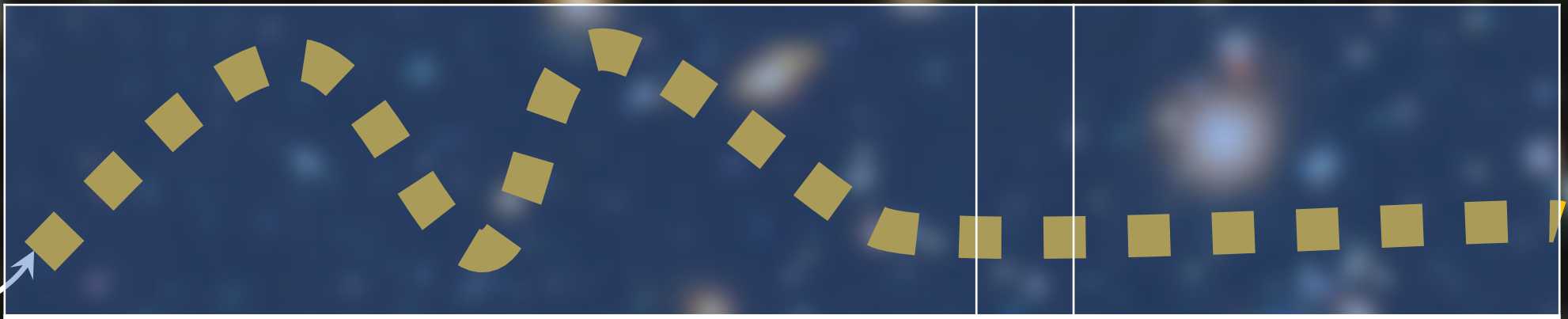
fd

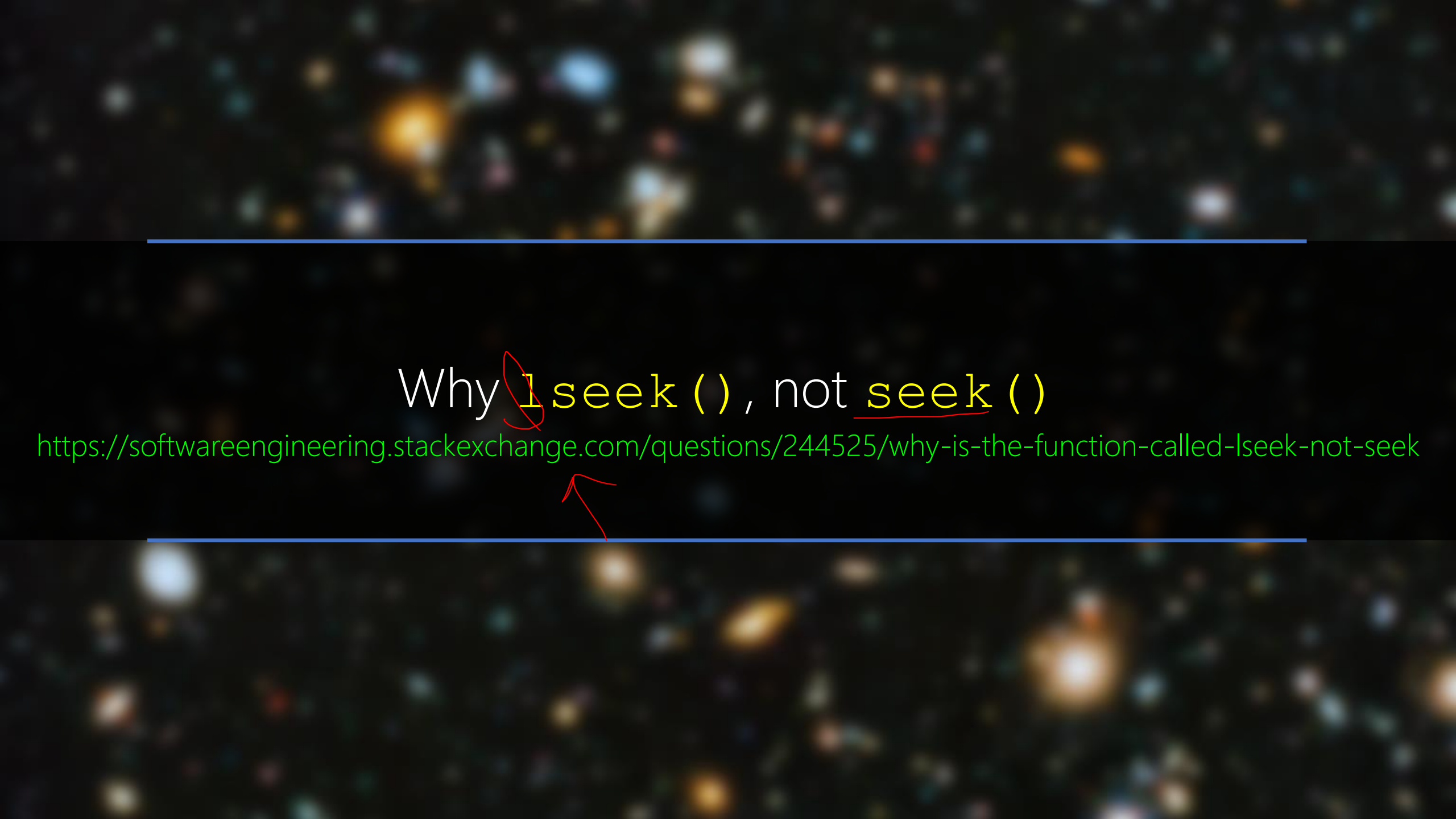


# lseek

A Better Use Case: Binary Search in Sorted Elements

fd





---

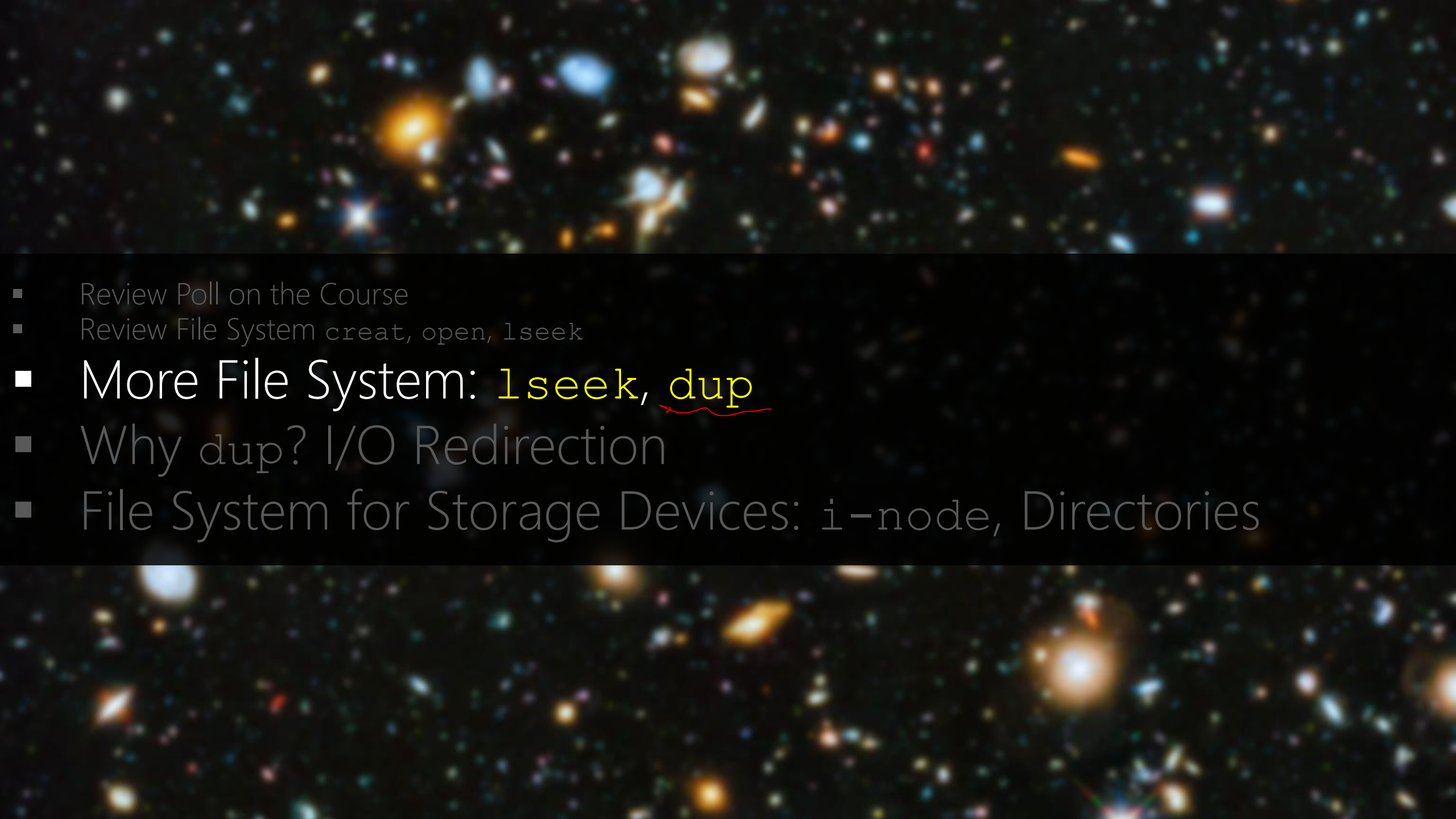
Why lseek(), not seek()

<https://softwareengineering.stackexchange.com/questions/244525/why-is-the-function-called-lseek-not-seek>

---





- 
- Review Poll on the Course
  - Review File System `creat`, `open`, `lseek`
  - More File System: `lseek`, `dup`
  - Why `dup`? I/O Redirection
  - File System for Storage Devices: `i-node`, Directories



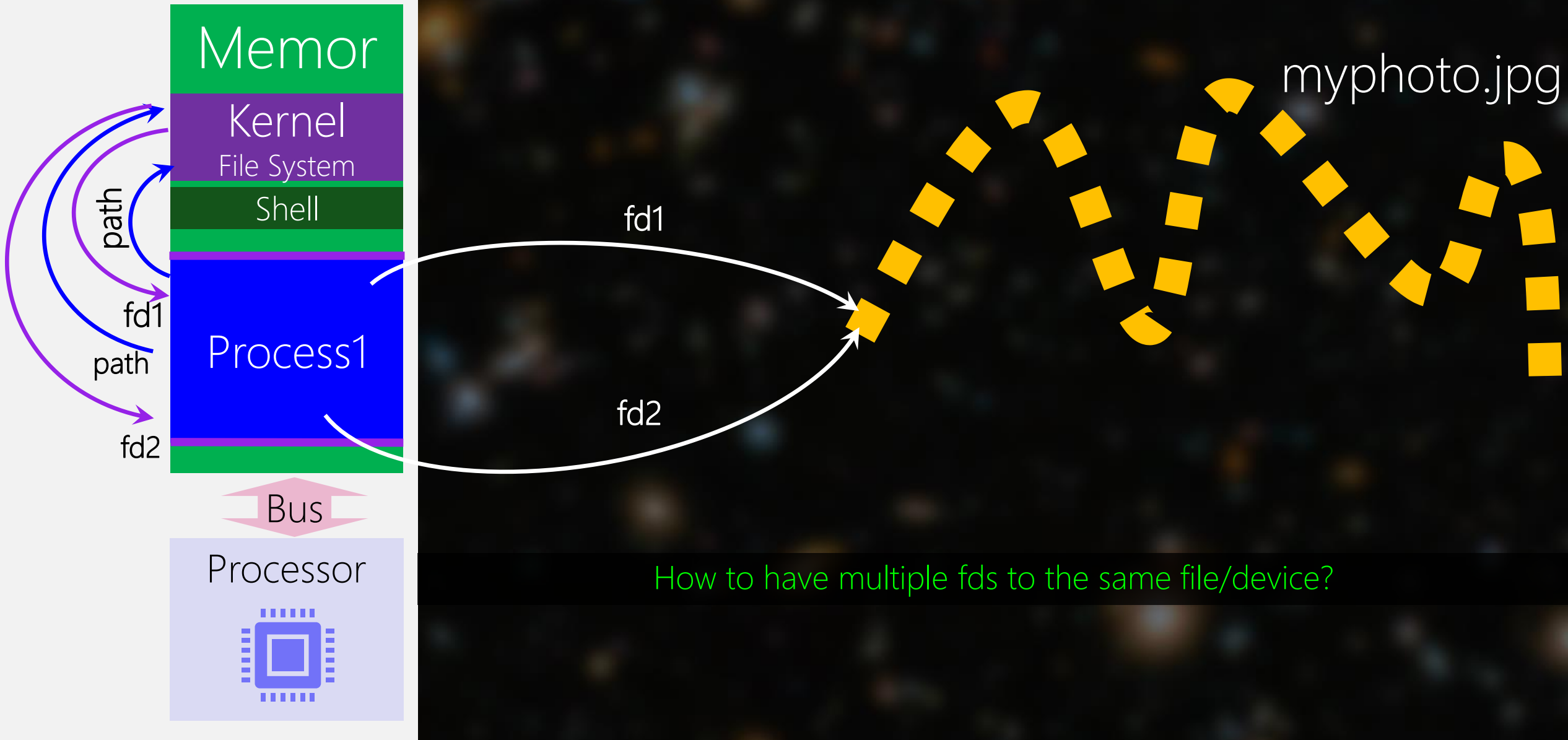
---

dup and dup2

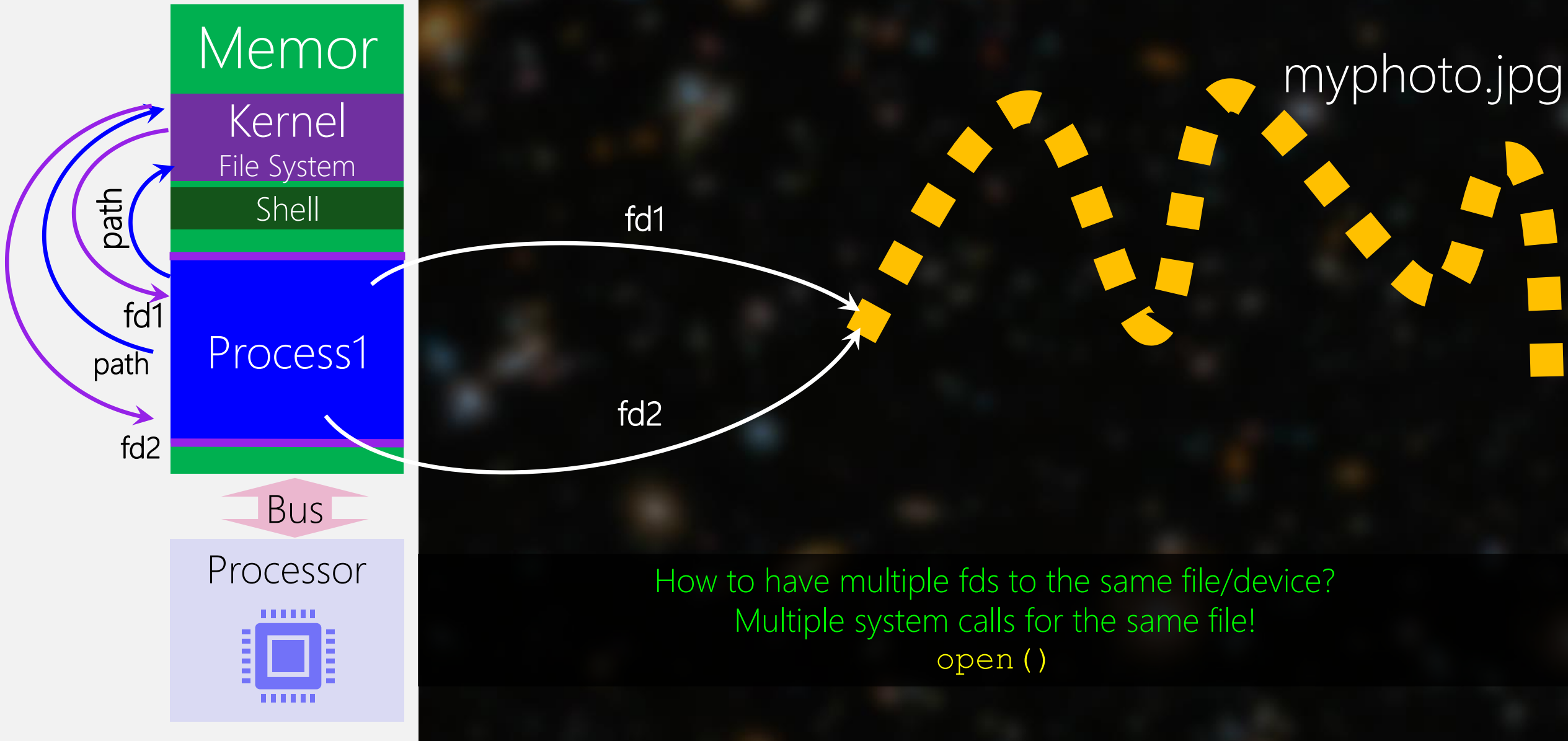
---

File System: High-Level

# Computer



# Computer





```

#include <fcntl.h>
#include <unistd.h>

#include <stdio.h>
void main(void) {

    int fd1 = open("test.txt", O_RDONLY | O_CREAT, S_IRUSR | S_IWUSR);
    if (fd1 == -1) {
        printf("error in opening file with 1st fd!");
    }
    printf("opening file with 1st fd: %d\n", fd1);
    int fd2 = open("test.txt", O_RDONLY | O_CREAT, S_IRUSR | S_IWUSR);
    if (fd2 == -1) {
        printf("error in opening file with 2nd fd!");
    }
    printf("opening file with 2nd fd: %d\n", fd2);
    int fd3 = open("test.txt", O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR);
    if (fd3 == -1) {
        printf("error in opening file with 3rd fd!");
    }
    printf("opening file with 3rd fd: %d\n", fd3);

}

```

hfani@bravo:~/comp2560\_f2021\$ cc duplicate\_open.c -o duplicate\_open

hfani@bravo:~/comp2560\_f2021\$ ./duplicate\_open

opening file with 1st fd: 3

opening file with 2nd fd: 4

opening file with 3rd fd: 5

## dup

```
#include <unistd.h>  
int dup(int fd);
```

the lowest-numbered available file descriptor to the same file if OK, -1 on error

## dup vs. multiple open

 open () : each returned fd has its own properties and current offset

fd1 = open ("test.txt", O\_RDONLY)

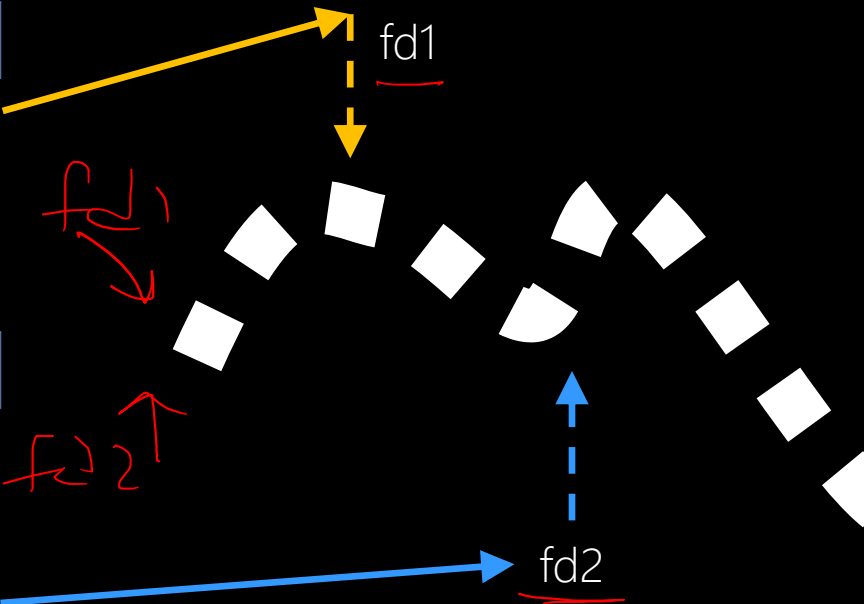
fd2 = open ("test.txt", O\_WRONLY)

Kernel  
File System

Process1

File Descriptors	File Pointer
fd1	
fd2	
...	

File Pointer
current file offset
other flags RD
pointer to first byte
File Pointer
current file offset
other flags WR
pointer to first byte





## dup vs. multiple open

`dup ()` : each returned fd has the same properties and current offset  
`fd2 = dup (fd1)`


Kernel  
File System

Process1

File Descriptors	File Pointer
fd1	
fd2	
...	

File Pointer
current file offset
other flags
pointer to first byte

fd1 fd2

- 
- Review Poll on the Course
  - Review File System `creat`, `open`, `lseek`
  - More File System: `lseek`, `dup`
  - Why `dup`? I/O Redirection
  - File System for Storage Devices: `i-node`, Directories

---

dup

Why do we duplicate fd?

---

I/O Redirection

STDIO ↔ File

STDERR → STDOUT



The image features a cosmic background of distant galaxies, appearing as colorful, out-of-focus spots of light in shades of blue, yellow, and white against a dark space. A solid black horizontal bar spans the width of the image, positioned centrally. Two thin, light blue horizontal lines are located just above and just below the black bar, extending across most of the image width.

Story

Write a program that

- accept two numbers from keyboard
- prints out sum in monitor

Write a program that

- accept two numbers from a file
- prints out sum in monitor

Write a program that

- accept two numbers from a keyboard
- prints out sum in a file

Write a program that

- accept two numbers from a file
- prints out sum in a file



Write a program that

- accept two numbers from the standard (default) input device
- prints out sum in the standard (default) output device



Write a program that

- accept two numbers from the STDIN
- prints out sum in the STDOUT





Write a program that

- accept two numbers from the STDIN\_FILENO
- prints out sum in the STDOUT\_FILENO

In UNIX, all devices are files!

```
#include <fcntl.h>
#include <unistd.h>

#include <stdio.h>
#include <stdlib.h>
void main(void) {
    //int fd1 = open(STDIN_FILENO, O_RDONLY);
    //if (fd1 == -1) {
    //    printf("error in opening stdin device!");
    //}
    char buf[100];

    read(STDIN_FILENO, buf, 100);
    int x = atoi(buf);
    read(STDIN_FILENO, buf, 100);
    int y = atoi(buf);

    //int fd2 = open(STDOUT_FILENO, O_WRONLY);
    //if (fd2 == -1) {
    //    printf("error in opening stdout device!");
    //}
    //printf("opening stdout with fd: %d\n", fd2);

    sprintf(buf, "%d\n", x + y);
    write(STDOUT_FILENO, buf, sizeof(int));
}
```

Reading from standard input

Writing to standard output

```
hfani@bravo:~/comp2560_f2021$ cc stdin_out.c -o stdin_out
hfani@bravo:~/comp2560_f2021$ ./stdin_out
```

21  
23  
44

Reading from standard input

Writing to standard output

Standard input and output is the same!  
Virtual Terminal: TeleTYpe (tty)



Write a program that

- accept two numbers from the 0
- prints out sum in the 1

In UNIX, fd of standard  
devices are predefined!



```
#include <fcntl.h>
#include <unistd.h>

#include <stdio.h>
#include <stdlib.h>
void main(void) {

    //int fd1 = open(STDIN_FILENO, O_RDONLY);
    //if (fd1 == -1){
    //    printf("error in opening stdin device!");
    //}
    char buf[100];

    read(0, buf, 100);
    int x = atoi(buf);
    read(0, buf, 100);
    int y = atoi(buf);

    //int fd2 = open(STDOUT_FILENO, O_WRONLY);
    //if (fd2 == -1){
    //    printf("error in opening stdout device!");
    //}
    //printf("opening stdout with fd: %d\n", fd2);

    sprintf(buf, "%d\n", x + y);
    write(1, buf, sizeof(int));

}
```

Reading from standard input



Writing to standard output



```
hfani@bravo:~/comp2560_f2021$ cc stdin_out_01.c -o stdin_out_01
hfani@bravo:~/comp2560_f2021$ ./stdin_out_01
```

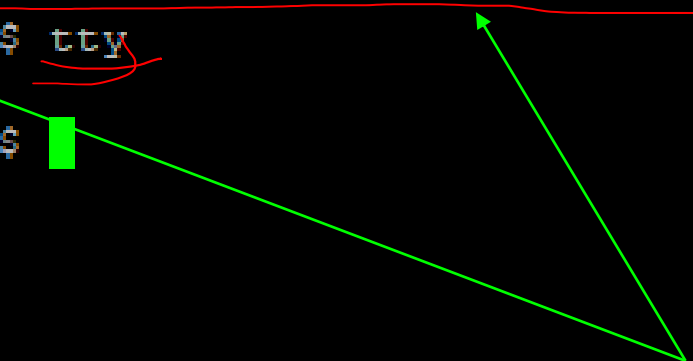
```
23
24
47
```

Reading from standard input

Writing to standard output

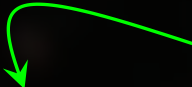
Standard input and output is the same!  
Virtual Terminal: TeleTYpe (tty)

```
hfani@bravo:~$ ls /dev/pts
0  1  10  11  12  13  14  15  18  2  3  4  5  6  7  8  9  ptmx
hfani@bravo:~$ tty
/dev/pts/18
hfani@bravo:~$
```



My Virtual Terminal: TeleTYpe (tty)

---

`#include stdio.h`  Reminder: the library routine

`fscanf() reads from STDIN_FILENO` 0

`fprintf() writes to STDOUT_FILENO` 1

---





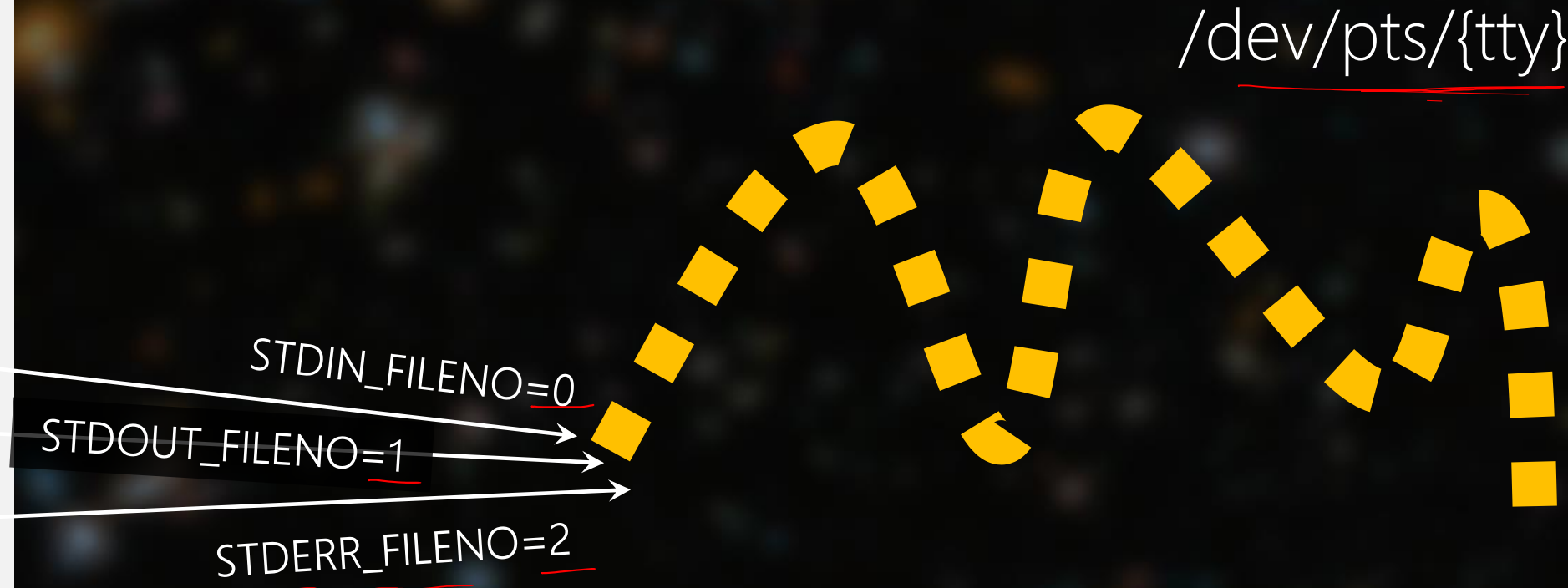
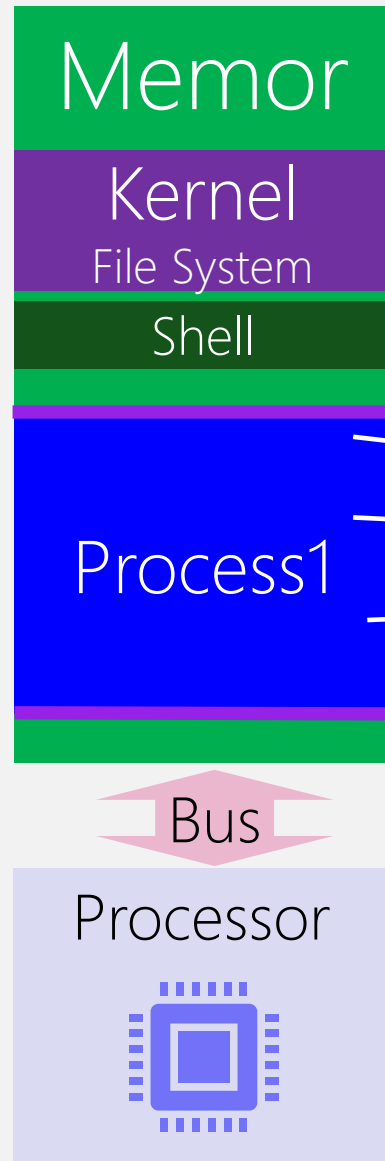
---

`read()` or `write()` without `open()`!

Who opened standard input and output devices?

---

# Computer



When Shell bootstraps a program, it automatically opens three fds for the program (process):

STDIN\_FILENO = 0 : O\_RDONLY  
STDOUT\_FILENO = 1 : O\_WRONLY  
STDERR\_FILENO = 2 : O\_WRONLY

```
#include <fcntl.h>
#include <unistd.h>
void main(void){

char buf_rd[20];

int fd_rd = open("/dev/fd/0" O_RDONLY);
int res = read(fd_rd, buf_rd, 10);
int fd_wr = open("/dev/fd/1", O_WRONLY);
write(fd_wr, buf_rd, 10);
```

```
#include <fcntl.h>
#include <unistd.h>
void main(void){

char buf_rd[20];

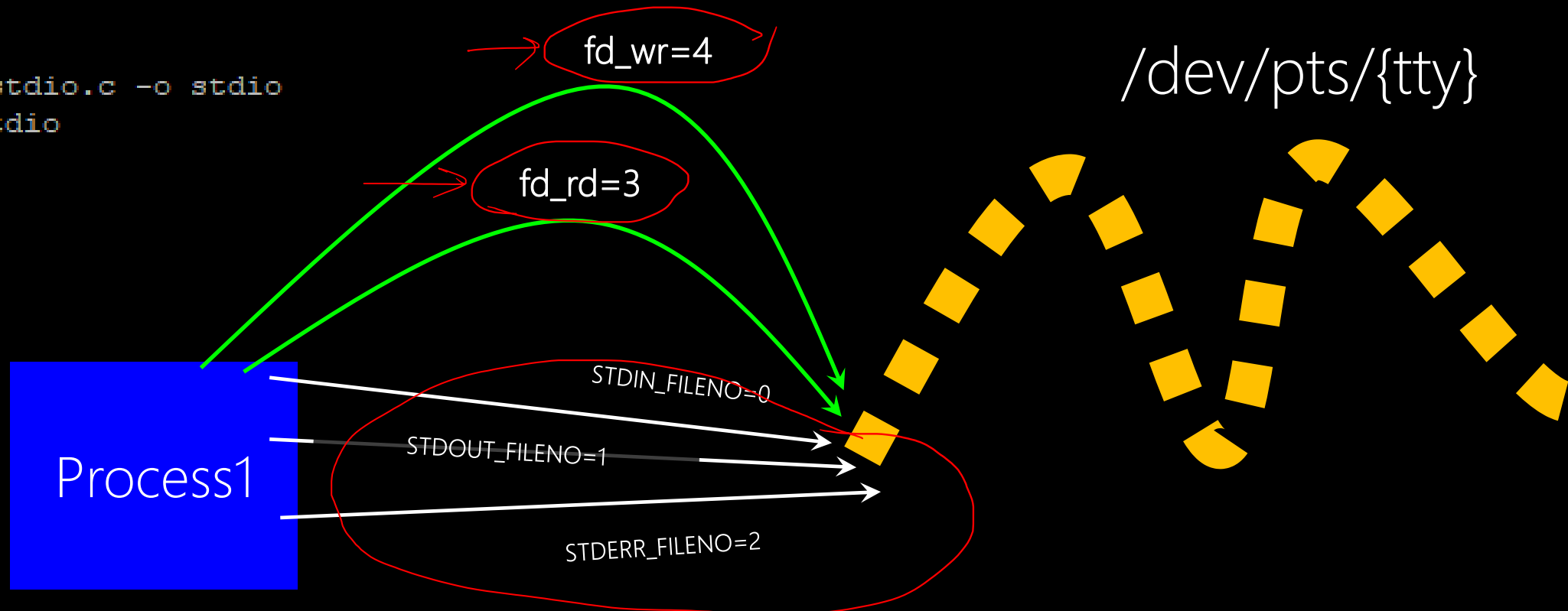
read(STDIN_FILENO, buf_rd, 10);
write(STDOUT_FILENO, buf_rd, 10);
```

```
#include <fcntl.h>
#include <unistd.h>
void main(void){

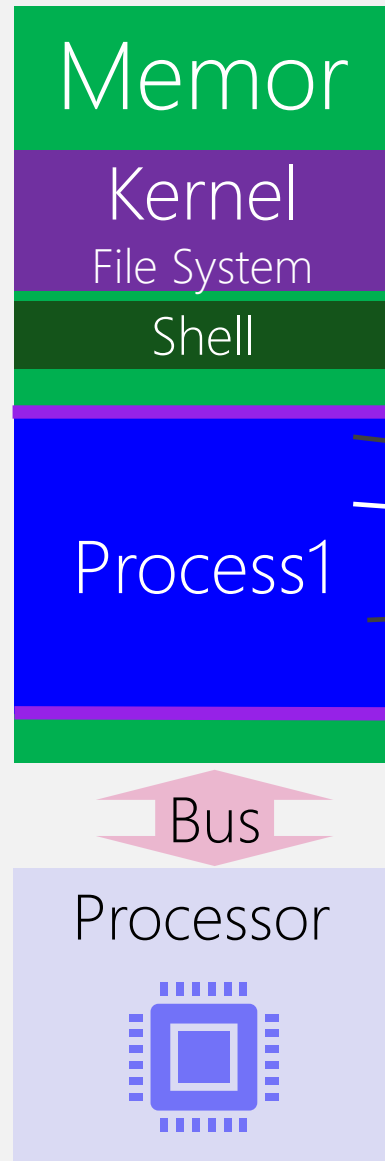
char buf_rd[20];

read(0, buf_rd, 10);
write(1, buf_rd, 10);
```

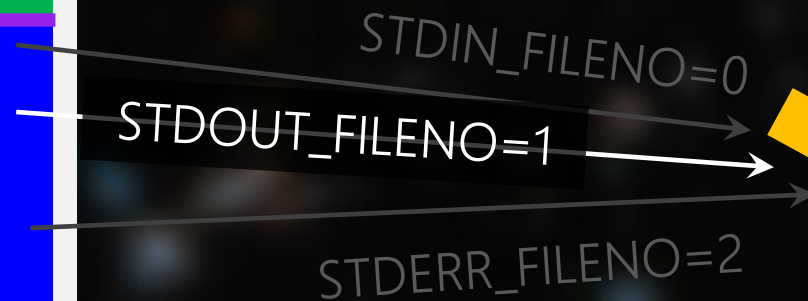
```
hfani@charlie:~$ cc stdio.c -o stdio
hfani@charlie:~$ ./stdio
comp2560
comp2560
hfani@charlie:~$
```



# Computer



/dev/pts/{tty}

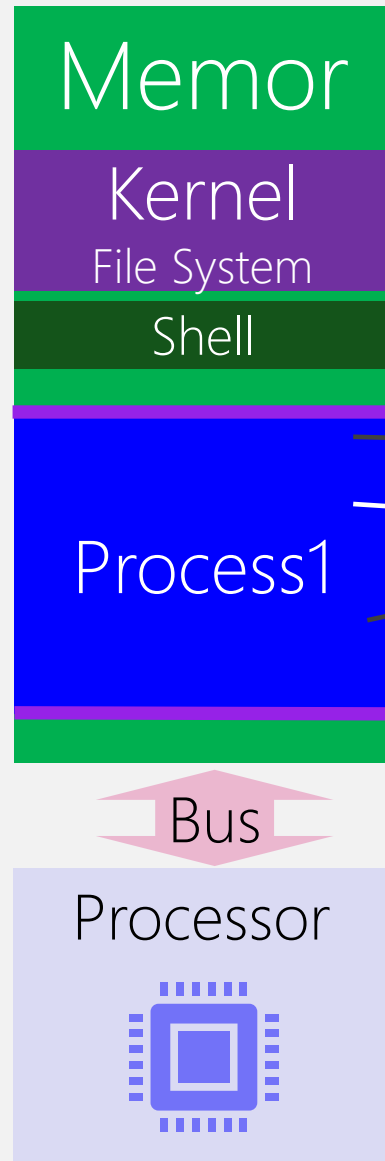


Now, we want to redirect output to a logging file ./log.txt

```
STDIN_FILENO = 0 : O_RDONLY  
STDOUT_FILENO = 1 : O_WRONLY  
STDERR_FILENO = 2 : O_WRONLY
```



# Computer



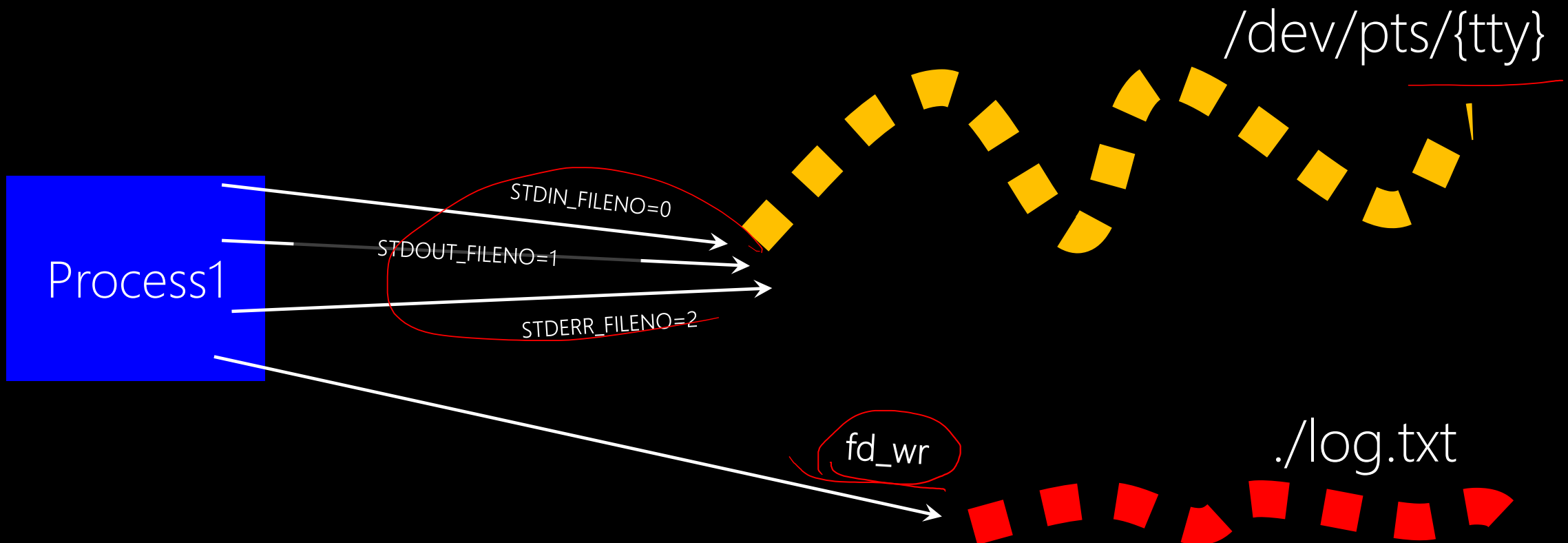
Now, we want to redirect output to a logging file ./log.txt

```
STDIN_FILENO = 0 : O_RDONLY  
STDOUT_FILENO = 1 : O_WRONLY  
STDERR_FILENO = 2 : O_WRONLY
```

```
#include <fcntl.h>
#include <unistd.h>
void main(void) {
```

Many lines of code that writes to STDOUT Like fprintf()  
But you want them in log.txt  
...

*write(fd\_wr)*



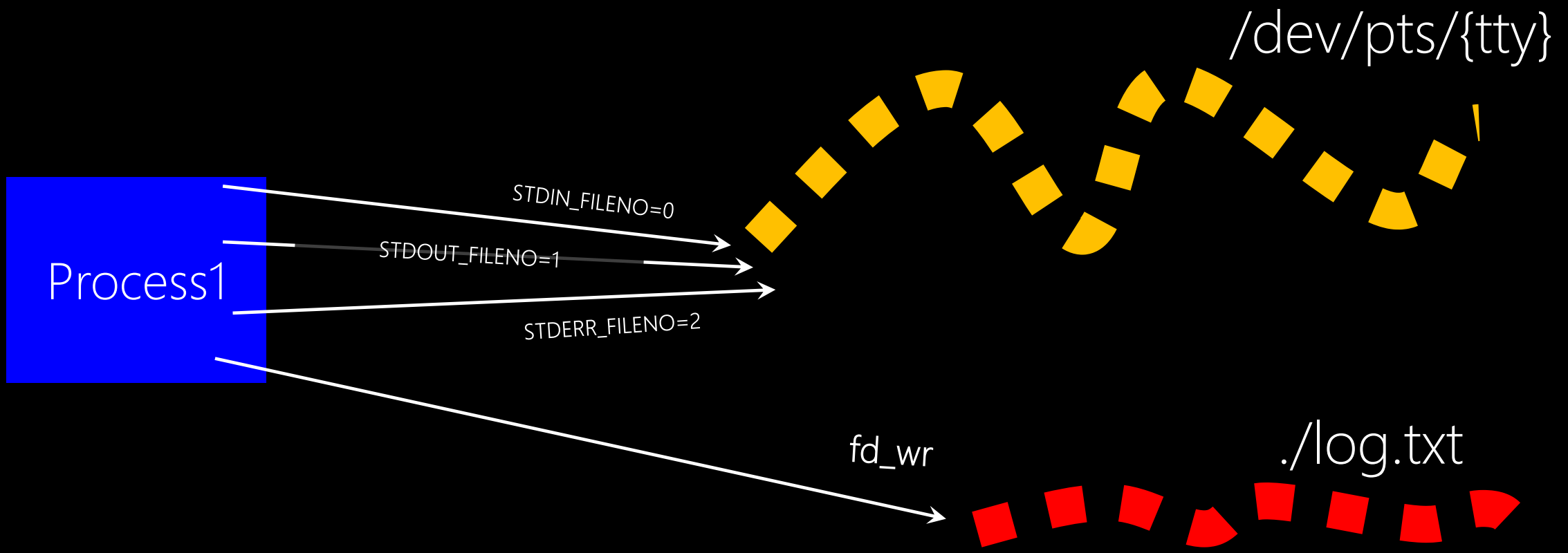
```
#include <fcntl.h>
#include <unistd.h>
void main(void){
    char buf_rd[20];

    int fd_wr = open("./log.txt", O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR);
```

...

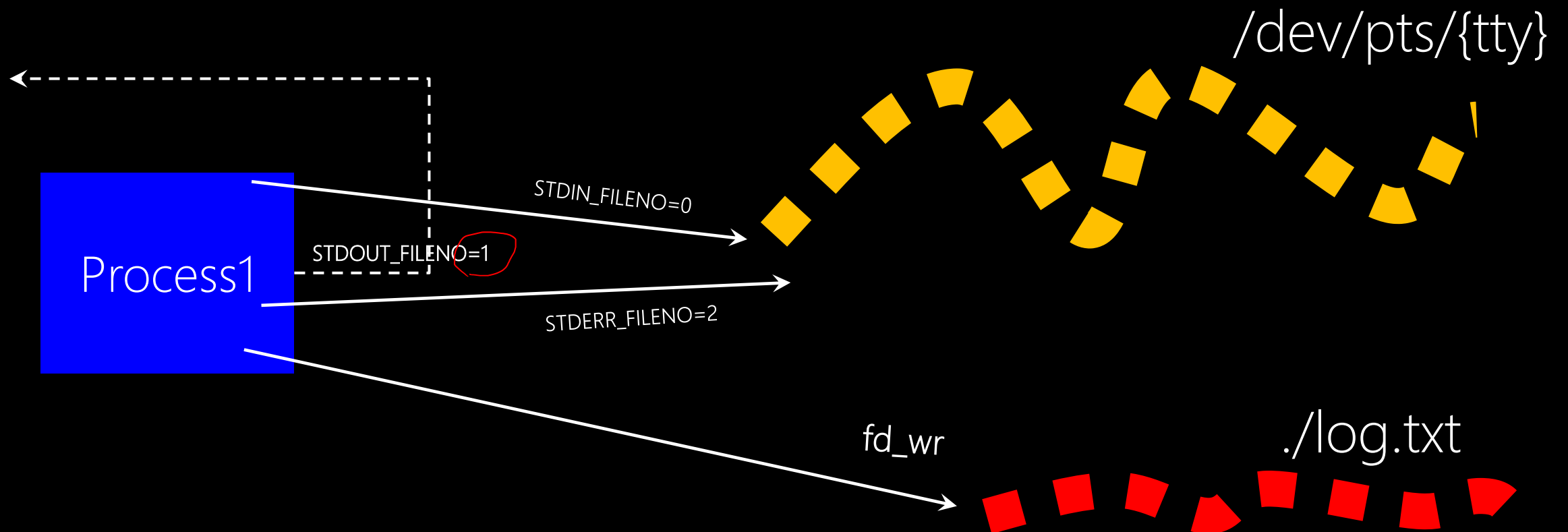
Many lines of code that writes to STDOUT Like fprintf()

...



```
#include <fcntl.h>
#include <unistd.h>
void main(void) {
    char buf_rd[20];

    int fd_wr = open("./log.txt", O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR);
    close(1);
    //now the fd with value 1 is free
```

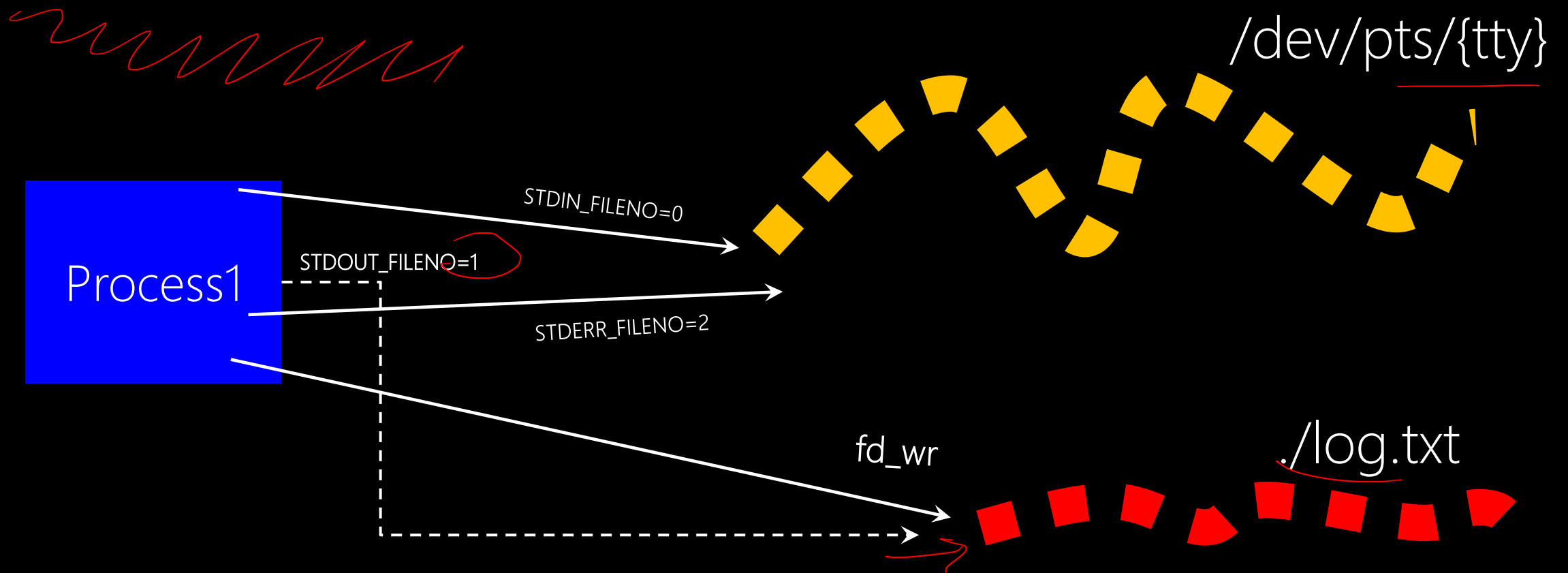




```
#include <fcntl.h>
#include <unistd.h>
void main(void) {
    char buf_rd[20];

    int fd_wr = open("../log.txt", O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR);
    close(1);
    //now the fd with value 1 is free
    //let's get it for our log file
    int new_fd = dup(fd_wr);
    //now the value of new_fd is 1
    //both fd_wr and new_fd are pointing to log.txt
```

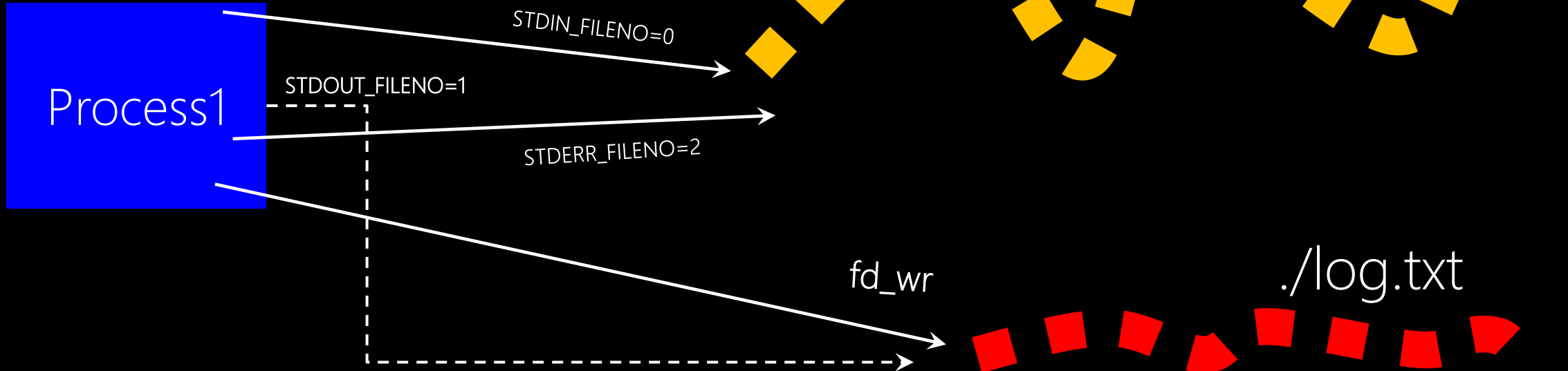
← the lowest-numbered available file descriptor



```
#include <fcntl.h>
#include <unistd.h>
void main(void){
    char buf_rd[20];

    int fd_wr = open("./log.txt", O_WRONLY | O_CREAT, S_IRUSR | S_IWUSR);
    close(1);
    //now the fd with value 1 is free
    //let's get it for our log file
    int new_fd = dup(fd_wr);
    //now the value of new_fd is 1
    //both fd_wr and new_fd are pointing to log.txt
```

```
read(STDIN_FILENO, buf_rd, 10);
write(STDOUT_FILENO, buf_rd, 10);
```



```
hfani@charlie:~$ cc stdio_redirection.c -o stdio_redirection
```

```
hfani@charlie:~$ ./stdio_redirection
```

```
hey again!
```

```
hfani@charlie:~$
```

```
hfani@charlie:~$ vi log.txt
```

```
hey again!
```

---

# Shell

\$ ./program > log.txt

---

We can ask the shell to do this redirection for us

*{program file} > {new destination for STDOUT\_FILENO}*





---

/

# Story

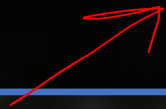
How about STDIN\_FILENO? STDERR\_FILENO?

---

# Story

What does this mean and what is the benefit?

fd = dup(0)



The background is a deep space image filled with numerous stars of various colors (white, yellow, orange, blue) against a black sky. Two horizontal blue lines are drawn across the image, one above and one below the central text.

dup2

At Home