Pintos project2-2

Before you start Project 2-2

- Make a new branch named “project2-2” and do your work at that branch.
  - git branch project2-2
  - git checkout project2-2

- **Without finishing 2-1, you can’t do 2-2. Please finish 2-1 first**
- **Also, Make commits frequently!**

- **DUE**
  - **5/19(Monday)**

Pre-work

- Read pintos manual
- 3. Project 2: User Programs
  - https://web.stanford.edu/class/cs140/projects/pintos/pintos_3.html#SEC32

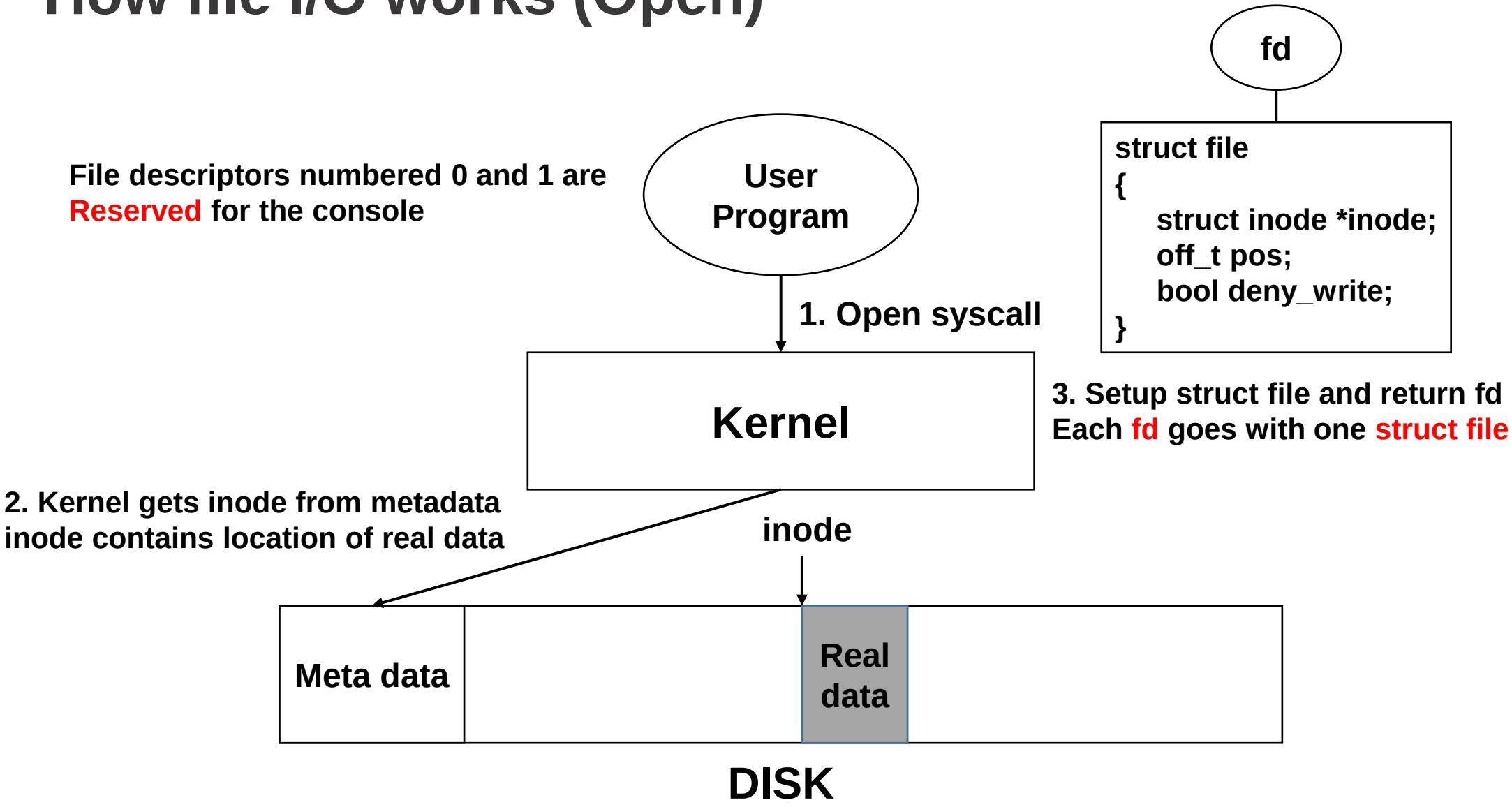
- Check your environment
  - $ cd ~/uni{student_id}/project2-2/pintos/src/userprog
  - $ make
  - $ cd build
  - $ make check

Goal for project 2-2

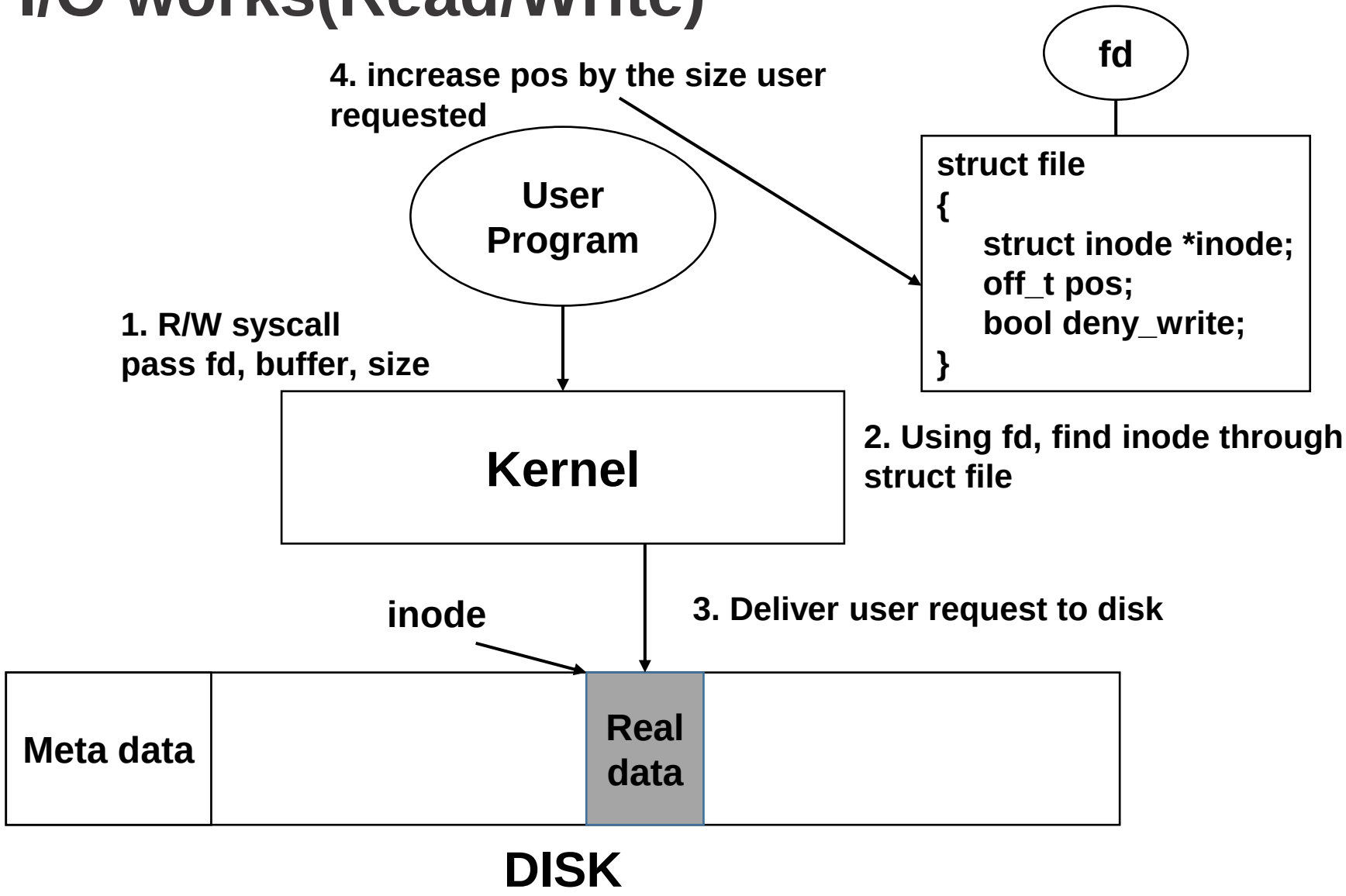
- **System call implementation**
  - read, write, exec, wait, remove, filesize, seek, tell

How file I/O works (Open)

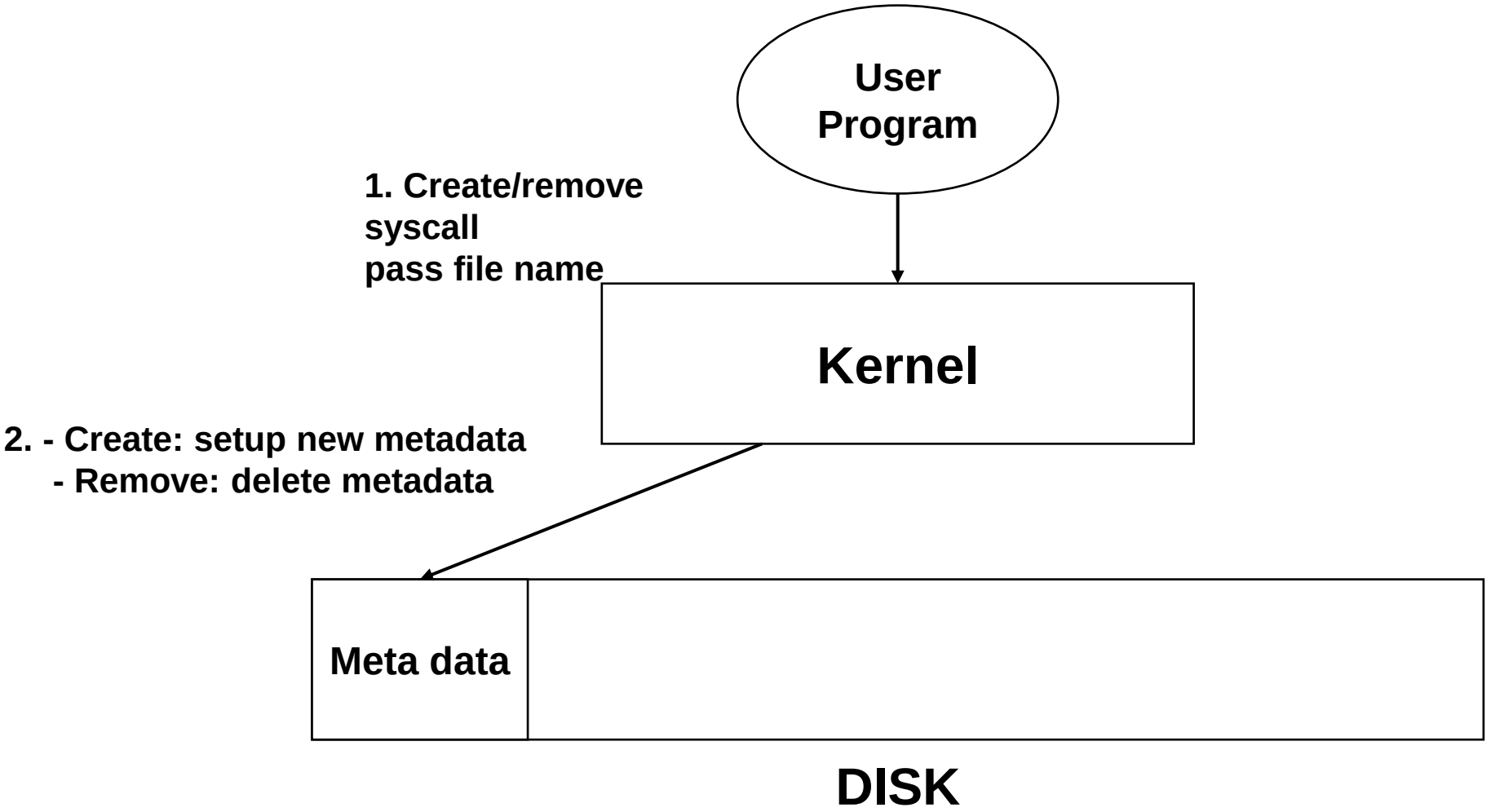
File descriptors numbered 0 and 1 are Reserved for the console



How file I/O works(Read/Write)



How file I/O works(Create/Remove)



What to implement

- int read (int *fd,* void **buffer*, unsigned *size*)
  - Read the file opened with *fd* to *buffer*
  - Return the number of bytes read, or -1 on fail
  - Fd 0 reads from the keyboard using input_getc()

- int write (int *fd,* const void **buffer*, unsigned *size*)
  - Fd 1 writes to console: call putbuf(*buffer, size*) and return *size,*
    It should be implemented already for pintos 2-1.
  - Else, write *size* bytes from *buffer* to the file opened with *fd*, return *size*

What to implement

- bool remove (const char *_file_)
  - Remove a file with name, *file*
  - If a file is open when it is removed, its blocks are not deallocated
    it may still be accessed by any threads
    that have it open, until the last one closes it

- pid_t exec (const char *_cmd_line)_
  - Executes *cmd_line* (program name and additional arguments, if any)
  - On success, returns new process's program id
  - On fail, return -1

What to implement

- int wait (pid_t *pid*)
- Please re-implement wait() following below instruction
  - wait for child process *pid* and retrieve the child's exit status
  - the child can already be terminated: return immediately

  - If child was killed by kernel (failure), return -1
  - If *pid* is not the direct child, return -1
    - even when *pid* is "grandson" of the caller
  - If wait was called more than twice, return -1

One thing for pintos project

- Test files(.c files) are located at src/tests/userprog folder, if you have difficulties for some test cases, try to see how they test
- Ex) *alarm-negative.c* in *src/tests/threads*

```
/* Tests timer_sleep(-100).  Only requirement is that it not crash. */

#include <stdio.h>
#include "tests/threads/tests.h"
#include "threads/malloc.h"
#include "threads/synch.h"
#include "threads/thread.h"
#include "devices/timer.h"

void
test_alarm_negative (void)
{
  timer_sleep (-100);
  pass ();
}
```

You have to pass

- **All tests**

Grading

- Code (pintos grade) : 90%
- Design document : 10%
  - Answer questions:
    - B1, B3, B4, B5, B6, B7, B8, B9, B10, B11

About questions

- We will not answer the question about detailed implementation and code, etc.
- Please ask questions only related to environment and high-level design of project.
- Please read pintos manual carefully.
- If you're facing an issue that just can't be resolved, contact me at **ogus05@unist.ac.kr**