

---

# Homework 5. JsonFS using FUSE

21800637 Jooyoung Jang

---

## Objective

---

Writing a FUSE program that constructs a userlevel file system based on the structure and data defined in a JSON file

---

# Ingredients

```
[
  {
    "inode": 0,
    "type": "dir",
    "entries":
    [
      { "name": "hello", "inode": 1},
      { "name": "d1", "inode": 2}
    ]
  },
  {
    "inode": 1,
    "type": "reg",
    "data": "Hello World!"
  },
  {
    "inode": 2,
    "type": "dir",
    "entries":
    [
      { "name": "d2", "inode": 3}
    ]
  },
]
```

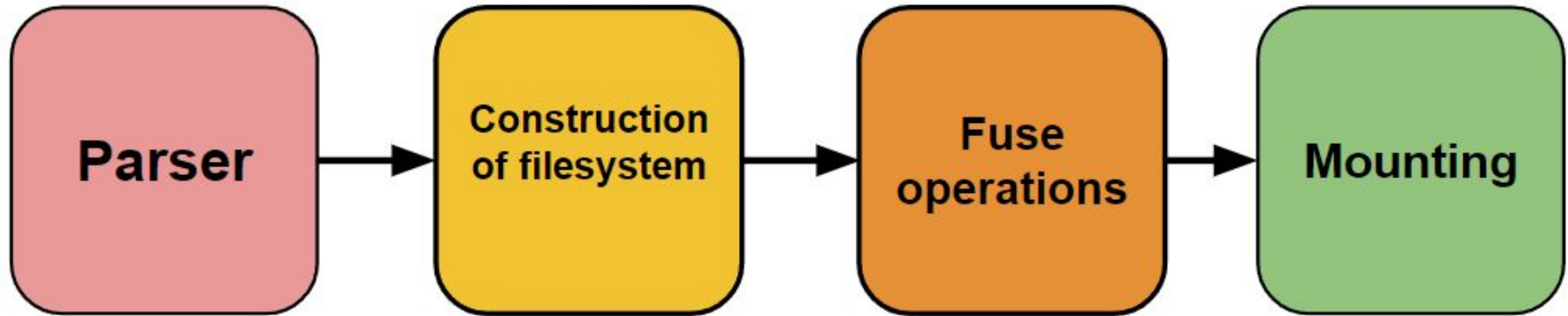
**fs.json**

```
1  /* taken from https://github.com/fntlnz/fuse-example, and modified
2
3
4  #define FUSE_USE_VERSION 26
5  #define MAX_FILE_BYTES 4098
6  #define MAX_FILES 16
7  #define TOTFILES 128
8
9  #include <stdio.h>
10 #include <fuse.h>
11 #include <string.h>
12 #include <errno.h>
13 #include <json.h>
14
15 /*typedef struct jsonfd_ent{
16     char * name;
17     int inode;
18 }jsonfd_ent*/
19
20
21
22
23
24 static const char *filecontent = "I'm the content of the only file";
25
26 static int getattr_callback(const char *path, struct stat *stbuf)
27 {
28     memset(stbuf, 0, sizeof(struct stat));
```

**fuse-example.c**

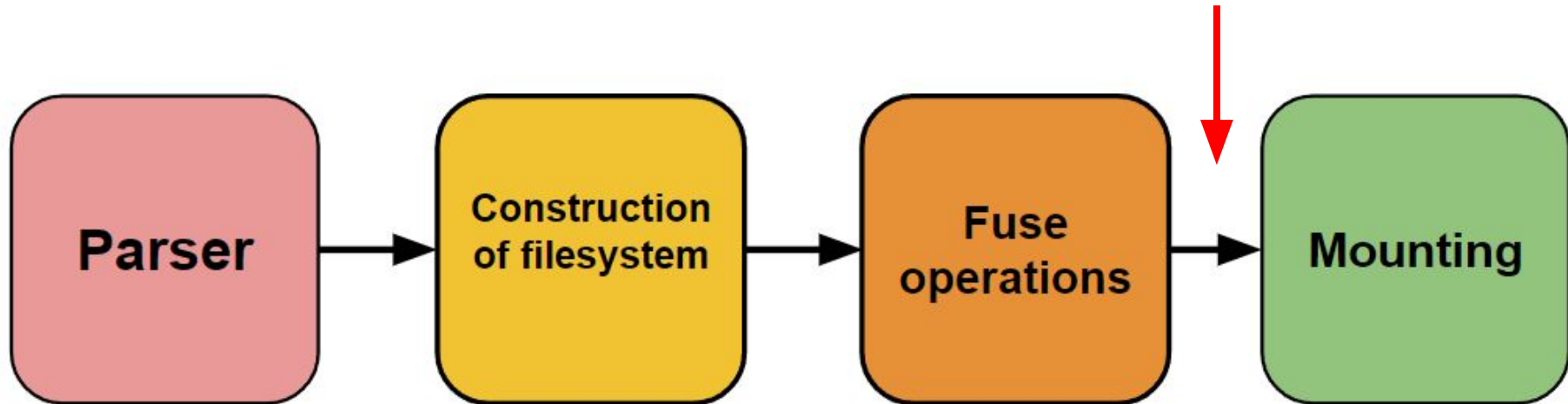
# Approach

---



## How far?

---



# What I have done

---

```
● s21800637@peace:~/HW/HW5$ ./run.sh
++ mkdir test
++ ./jsonfsx fs.json ./test
Created file object with inode 0, type dir
Entries:
inode: 1, name: hello
inode: 2, name: d1
Created file object with inode 1, type reg
Created file object with inode 2, type dir
Entries:
inode: 3, name: d2
Created file object with inode 3, type dir
Entries:
inode: 4, name: bye
inode: 1, name: hello
Created file object with inode 4, type reg
Created file object with inode 5, type reg
```

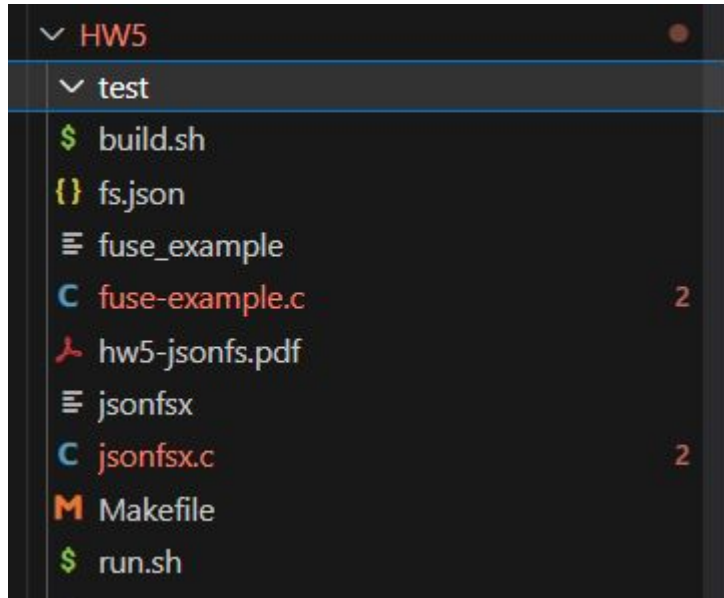
```
✓ static struct fuse_operations jsonfs_oper = {
    .getattr = getattr_callback,
    .readdir = readdir_callback,
    .mkdir = mkdir_callback,
    .rmdir = rmdir_callback,
    .create = create_callback,
    .unlink = unlink_callback,
    .write = write_callback,
    .link = link_callback,
    .open = open_callback,
    .read = read_callback,
    .release = release_callback,
    .rename = rename_callback,
};
```

Parsing

fuse functions

# What I have done

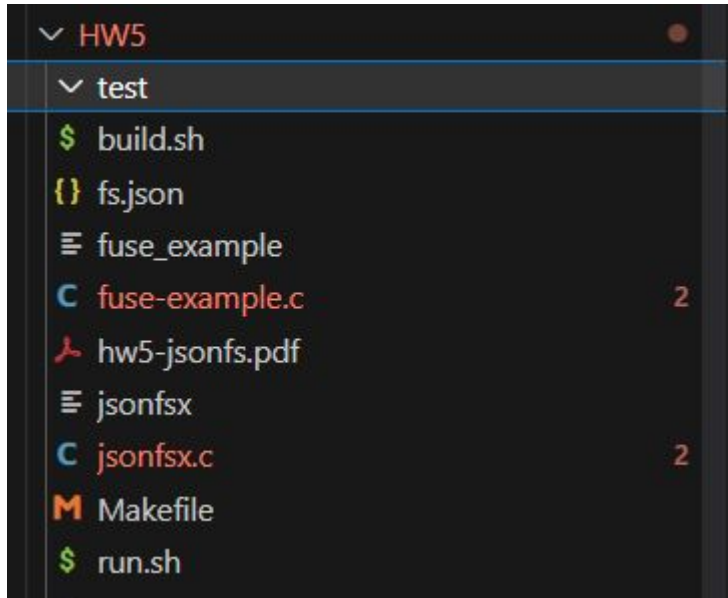
---



Some part of mounting

# What I couldn't figure out

---



```
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <pthread.h>
#include <json-c/json.h>
#include <json-c/json_object.h>
```

Some part of mounting

Concurrency



# Disucussion

---

- **Why couldn't I mount the whole file system structure?**
  - **The issue regarding the permission of files and directories**
-

# Improvements

---

- **Concurrency**
- **Better Data Structure**
- **Full understanding of fuse functionality**