운영체제론 Project-2

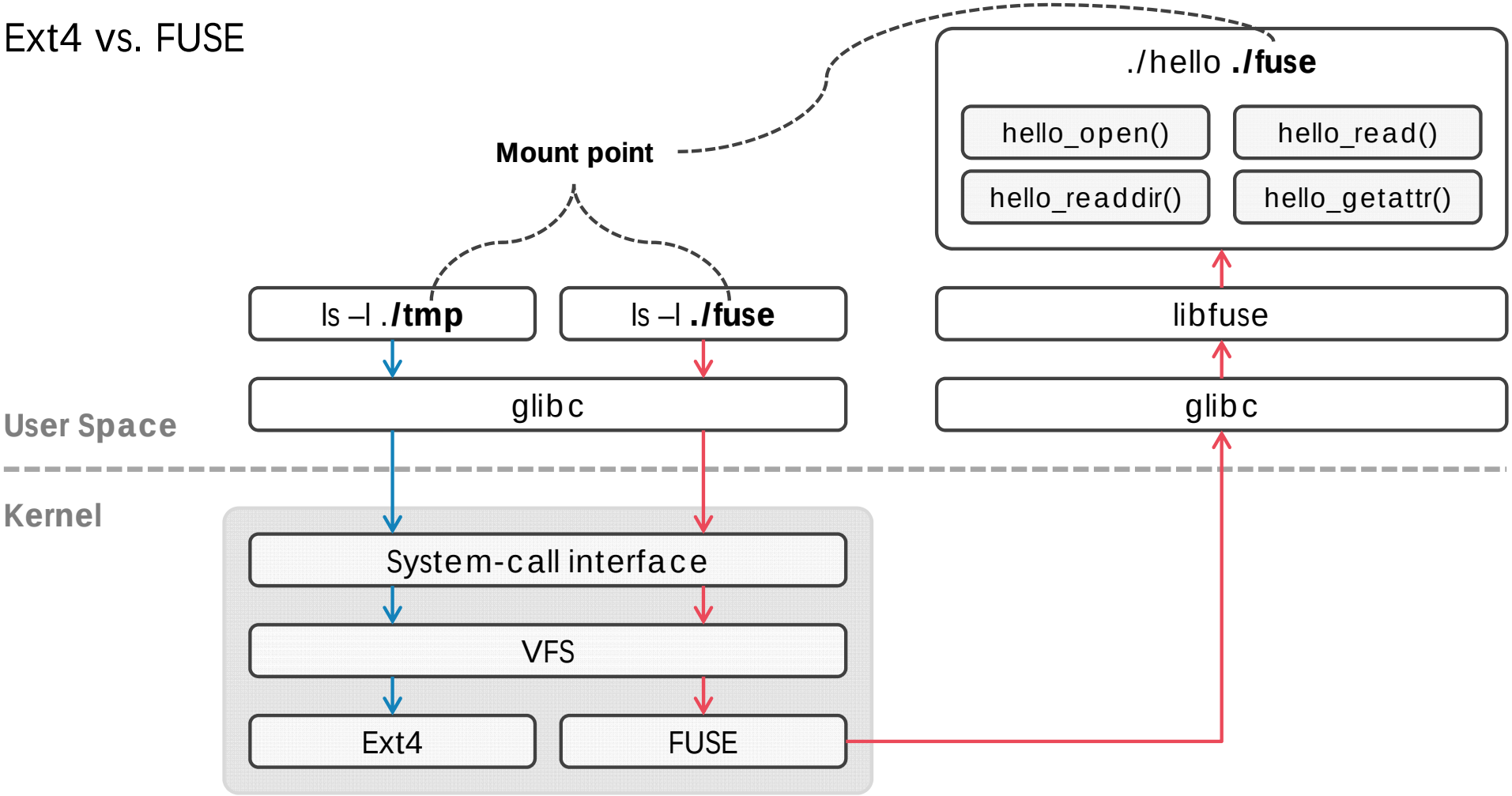
FUSE-based File System Implementation

2019.05.23

Contents

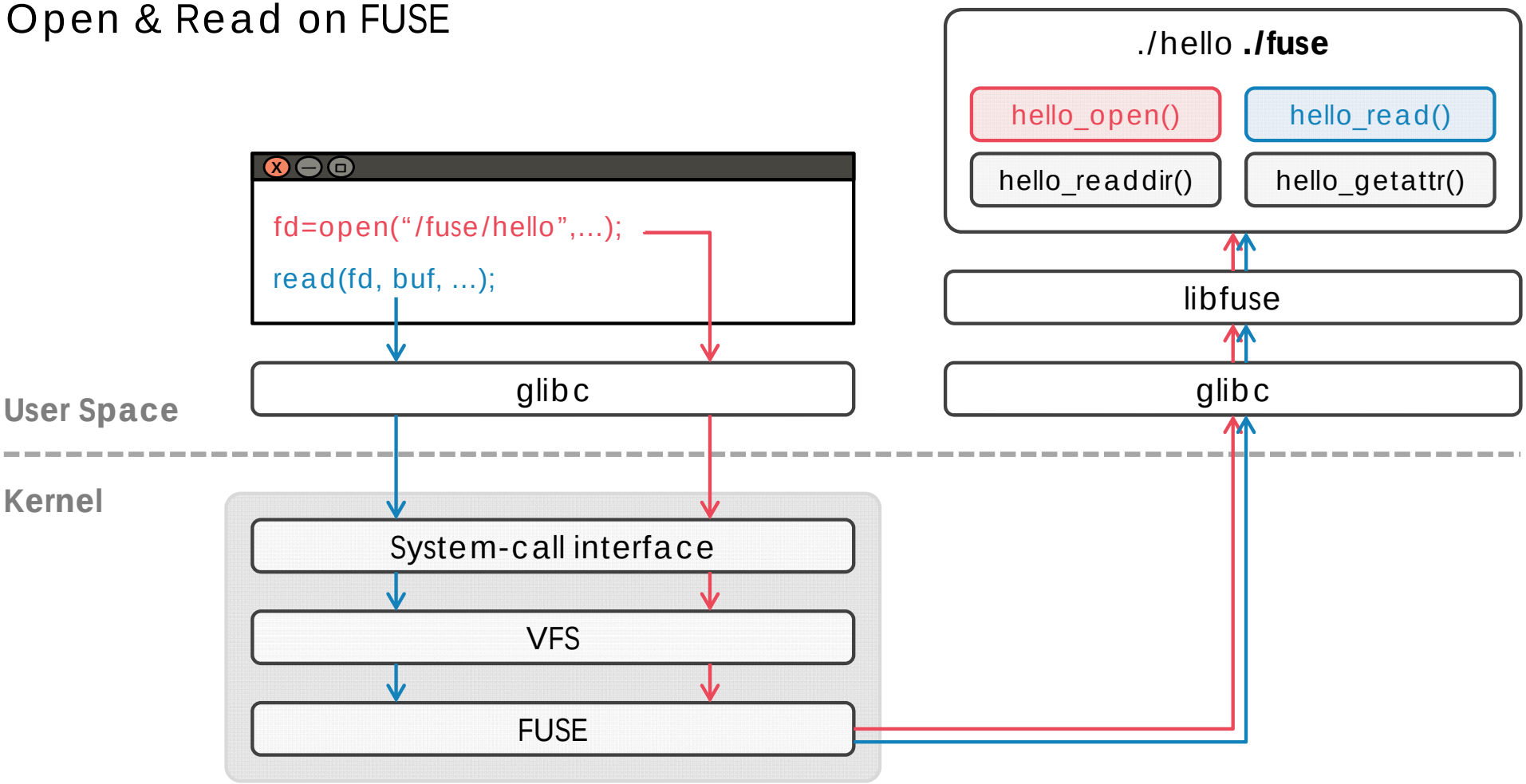
FUSE Introduction

- FUSE (Filesystem in USErspace)
  - Ext4 vs. FUSE



FUSE Introduction

- FUSE (Filesystem in USErspace)
  - Open & Read on FUSE



FUSE Operations

- File/Directory Management
  - File/Directory Descriptor (Tree)

- Fuse Operations
  - File Operations
  - Directory Operations
  - Metadata Operations
  - Other Operations

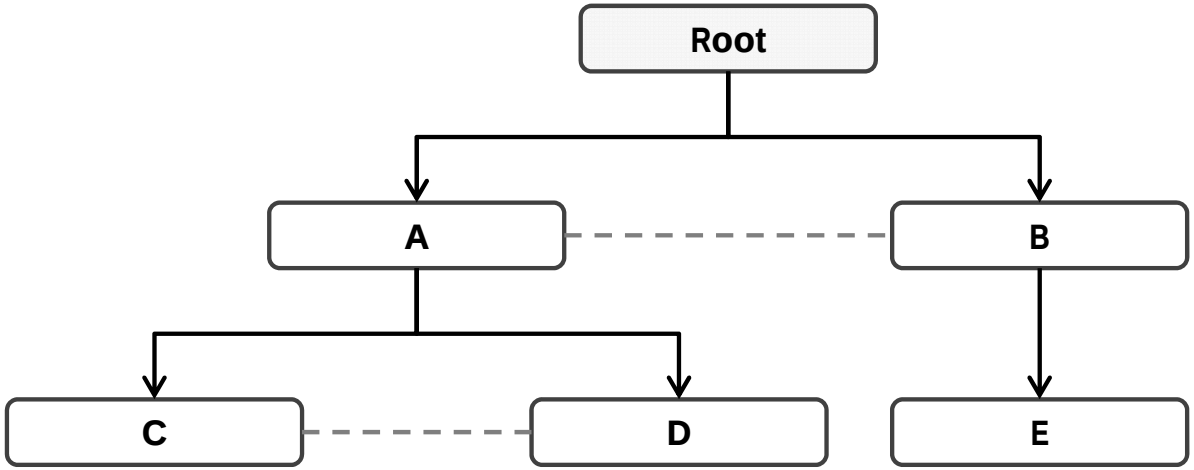
- Useful Errors in FUSE
- FUSE context

FUSE Operations

- File/Directory Descriptor
  - size: size in bytes
  - mode: type and permissions
  - uid: owner id
  - gid: group id
  - atime: access time (often fudged)
  - mtime: modification time
  - ctime: metadata change time
  - data: file data

FUSE Operations

- File/Directory Descriptor (Tree)
  - next: next node pointer
  - child: child node pointer
  - parents: parents node pointer



FUSE Operations

- File Operations
  - mknod(path, mode, dev): create a file (or device)
  - unlink(path): remove (delete) the give file, link, etc.
  - rename(old, new): move and/or rename a file
  - open(path, flags): open a file
  - read(path, length, offset, fi)
  - write(path, buf, offset, fi)
  - truncate(path, len, fi): cut off at length
  - flush(path, fi): one handle is closed
  - release(path, fi): file handle is completely closed (no errors)

FUSE Operations

- Directory Operations
  - readdir(path): yield directory entries for each file in the directory
  - mkdir(path, mode): create a directory
  - rmdir(path): delete an empty directory

- Metadata Operations
  - getattr(path): read metadata
  - chmod(path, mode): alter permissions
  - chown(path, uid, gid): alter ownership

FUSE Operations

- Other Operations
  - statfs(path)
  - fsdestroy()
  - create(path, flags, mode)
  - utimens(path, times)
  - readlink(path)
  - symlink(target, name)
  - link(target, name)
  - fsync(path, fdatasync, fi)
  - etc.

FUSE Operations

- fuse_operations [Link] https://libfuse.github.io/doxygen/structfuse__operations.html

```
int(*  getattr )(const char *, struct stat *, struct fuse_file_info *fi)
int(*  readlink )(const char *, char *, size_t)
int(*  mknod )(const char *, mode_t, dev_t)
int(*  mkdir )(const char *, mode_t)
int(*  unlink )(const char *)
int(*  rmdir )(const char *)
int(*  symlink )(const char *, const char *)
int(*  rename )(const char *, const char *, unsigned int flags)
int(*  link )(const char *, const char *)
int(*  chmod )(const char *, mode_t, struct fuse_file_info *fi)
int(*  chown )(const char *, uid_t, gid_t, struct fuse_file_info *fi)
int(*  truncate )(const char *, off_t, struct fuse_file_info *fi)
int(*  open )(const char *, struct fuse_file_info *)
int(*  read )(const char *, char *, size_t, off_t, struct fuse_file_info *)
int(*  write )(const char *, const char *, size_t, off_t, struct fuse_file_info *)
int(*  statfs )(const char *, struct statvfs *)
int(*  flush )(const char *, struct fuse_file_info *)
int(*  release )(const char *, struct fuse_file_info *)
int(*  fsync )(const char *, int, struct fuse_file_info *)
int(*  setxattr )(const char *, const char *, const char *, size_t, int)
int(*  getxattr )(const char *, const char *, char *, size_t)
int(*  listxattr )(const char *, char *, size_t)
int(*  removexattr )(const char *, const char *)
```

FUSE Operations

- fuse_operations [Link] https://libfuse.github.io/doxygen/structfuse__operations.html

```
int(*  opendir )(const char *, struct fuse_file_info *)
int(*  readdir )(const char *, void *, fuse_fill_dir_t, off_t, struct fuse_file_info *, enum fuse_readdir_flags)
int(*  releasedir )(const char *, struct fuse_file_info *)
int(*  fsyncdir )(const char *, int, struct fuse_file_info *)
void *(*  init )(struct fuse_conn_info *conn, struct fuse_config *cfg)
void(*  destroy )(void *private_data)
int(*  access )(const char *, int)
int(*  create )(const char *, mode_t, struct fuse_file_info *)
int(*  lock )(const char *, struct fuse_file_info *, int cmd, struct flock *)
int(*  utimens )(const char *, const struct timespec tv[2], struct fuse_file_info *fi)
int(*  bmap )(const char *, size_t blocksize, uint64_t *idx)
int(*  ioctl )(const char *, unsigned int cmd, void *arg, struct fuse_file_info *, unsigned int flags, void *data)
int(*  poll )(const char *, struct fuse_file_info *, struct fuse_pollhandle *ph, unsigned *reventsp)
int(*  write_buf )(const char *, struct fuse_bufvec *buf, off_t off, struct fuse_file_info *)
int(*  read_buf )(const char *, struct fuse_bufvec **bufp, size_t size, off_t off, struct fuse_file_info *)
int(*  flock )(const char *, struct fuse_file_info *, int op)
int(*  fallocate )(const char *, int, off_t, off_t, struct fuse_file_info *)
ssize_t(*  copy_file_range )(const char *path_in, struct fuse_file_info *fi_in, off_t offset_in, const char *path_out, struct fuse_file_info *fi_out, off_t offset_out, size_t size, int flags)
```

FUSE Operations

- Useful Errors in FUSE
  - Error Lists
    - → errno.ENOSYS: Function not implemented
    - → errno.EROFS: Read-only file system
    - → errno.EPERM: Operation not permitted
    - → errno.EACCES: Permission denied
    - → errno.ENOENT: No such file or directory
    - → errno.EIO: I/O error
    - → errno.EEXIST: File exists
    - → errno.ENOTDIR: Not a directory
    - → errno.EISDIR: Is a directory
    - → errno.ENOTEMPTY: Directory not empty

FUSE Operations

- Useful Errors in FUSE
  - Ex) Error Check

    → errno.ENOENT: No such file or directory

```c
static int hello_readdir(const char *path, void *buf, fuse_fill_dir_t filler,
                         off_t offset, struct fuse_file_info *fi)
{
        (void) offset;
        (void) fi;

        if (strcmp(path, "/") != 0)
                return -ENOENT;

        filler(buf, ".", NULL, 0);
        filler(buf, "..", NULL, 0);
        filler(buf, hello_path + 1, NULL, 0);

        return 0;
}
```

FUSE Operations

- FUSE Context
  - fuse_get_context() returns struct fuse_context with:
    - → uid: User ID of the calling process
    - → gid: Group ID of the calling process
    - → pid: Thread ID of the calling process
    - → private_data: Private filesystem data
    - → umask: Umask of the calling process
  - Useful for nonstandard permission models and other user-specific behavior

Sample

- Hello File System
  - A single file (Hello) can be read (without write/create/delete)
    [Link] https://github.com/fuse4x/fuse/blob/master/example/hello.c
  - ex) hello.c - main function

```c
#define FUSE_USE_VERSION 26

#include <fuse.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>

static const char *hello_str = "Hello World!\n";
static const char *hello_path = "/hello";
```

```c
static struct fuse_operations hello_oper = {
        .getattr        = hello_getattr,
        .readdir        = hello_readdir,
        .open           = hello_open,
        .read           = hello_read,
};

int main(int argc, char *argv[])
{
        return fuse_main(argc, argv, &hello_oper, NULL);
}
```

Sample

- Hello File System
  - ex) hello.c - Open function: Check file path and access rights to FILE

```c
static int hello_open(const char *path, struct fuse_file_info *fi)
{
        if (strcmp(path, hello_path) != 0)
                return -ENOENT;

        if ((fi->flags & 3) != O_RDONLY)
                return -EACCES;

        return 0;
}
```

  - ex) hello.c - Read function: Send corresponding data to file using file offset

```c
static int hello_read(const char *path, char *buf, size_t size, off_t offset,
                      struct fuse_file_info *fi)
{
        size_t len;
        (void) fi;
        if(strcmp(path, hello_path) != 0)
                return -ENOENT;

        len = strlen(hello_str);
        if (offset < len) {
                if (offset + size > len)
                        size = len - offset;
                memcpy(buf, hello_str + offset, size);
        } else
                size = 0;

        return size;
}
```

Sample

- Hello File System
  - ex) hello.c - Readdir function: send the list of files in the directory to buf

```
static int hello_readdir(const char *path, void *buf, fuse_fill_dir_t filler,
                         off_t offset, struct fuse_file_info *fi)
{
        (void) offset;
        (void) fi;

        if (strcmp(path, "/") != 0)
                return -ENOENT;

        filler(buf, ".", NULL, 0);
        filler(buf, "..", NULL, 0);
        filler(buf, hello_path + 1, NULL, 0);

        return 0;
}
```

Sample

- Hello File System
  - ex) hello.c - Getattr function: send the attribute of path file

```
static int hello_getattr(const char *path, struct stat *stbuf)
{
        int res = 0;

        memset(stbuf, 0, sizeof(struct stat));
        if (strcmp(path, "/") == 0) {
                stbuf->st_mode = S_IFDIR | 0755;
                stbuf->st_nlink = 2;
        } else if (strcmp(path, hello_path) == 0) {
                stbuf->st_mode = S_IFREG | 0444;
                stbuf->st_nlink = 1;
                stbuf->st_size = strlen(hello_str);
        } else
                res = -ENOENT;

        return res;
}
```

Demo

- Install the FUSE library on Ubuntu

```
sudo apt-get install libfuse-dev
```

  - [Link] https://github.com/libfuse/libfuse/

- Source code download (Optional)
  - [Link] https://github.com/fuse4x/fuse
  - [Link] https://github.com/fuse4x/fuse/blob/master/example/hello.c

```
minhozx@dclab-PowerEdge-R940:~/Workspace$ git clone https://github.com/fuse4x/fuse
Cloning into 'fuse'...
remote: Enumerating objects: 8129, done.
remote: Total 8129 (delta 0), reused 0 (delta 0), pack-reused 8129
Receiving objects: 100% (8129/8129), 2.11 MiB | 553.00 KiB/s, done.
Resolving deltas: 100% (6388/6388), done.
Checking connectivity... done.
minhozx@dclab-PowerEdge-R940:~/Workspace$ ls
fuse
minhozx@dclab-PowerEdge-R940:~/Workspace$ cp fuse/example/hello.c ./
minhozx@dclab-PowerEdge-R940:~/Workspace$ ls
fuse  hello.c
minhozx@dclab-PowerEdge-R940:~/Workspace$
```

Demo

- Compile

```
gcc -D_FILE_OFFSET_BITS=64 -o hello hello.c -lfuse
```

```
minhozx@dclab-PowerEdge-R940:~/Workspace$ gcc -D_FILE_OFFSET_BITS=64 -o hello hello.c -lfuse
minhozx@dclab-PowerEdge-R940:~/Workspace$ ls
fuse  hello  hello.c
minhozx@dclab-PowerEdge-R940:~/Workspace$
```

- Mount
  - ./hello [mount point]

```
./hello ./mnt
```

```
minhozx@dclab-PowerEdge-R940:~/Workspace$ gcc -D_FILE_OFFSET_BITS=64 -o hello hello.c -lfuse
minhozx@dclab-PowerEdge-R940:~/Workspace$ ls
fuse  hello  hello.c
minhozx@dclab-PowerEdge-R940:~/Workspace$ mkdir mnt
minhozx@dclab-PowerEdge-R940:~/Workspace$ ./hello ./mnt/
minhozx@dclab-PowerEdge-R940:~/Workspace$ mount | grep 'mnt'
/home/minhozx/Workspace/hello on /home/minhozx/Workspace/mnt type fuse.hello (rw,nosuid,nodev,relatime,user_id=1003,group_id=1003)
minhozx@dclab-PowerEdge-R940:~/Workspace$
```

Demo

- Reading a file
  - cat [file]

```
cat hello
```

```
minhozx@dclab-PowerEdge-R940:~/Workspace$ mount | grep 'mnt'
/home/minhozx/Workspace/hello on /home/minhozx/Workspace/mnt type fuse.hello (rw,nosuid,nodev,relatime,user_id=1003,group_id=1003)
minhozx@dclab-PowerEdge-R940:~/Workspace$ cd mnt/
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$ ls -al
total 4
drwxr-xr-x 2 root    root       0  1월  1  1970 .
drwxrwxr-x 4 minhozx minhozx 4096  5월 22 23:14 ..
-r--r--r-- 1 root    root      13  1월  1  1970 hello
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$ cat hello
Hello World!
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$
```

Demo

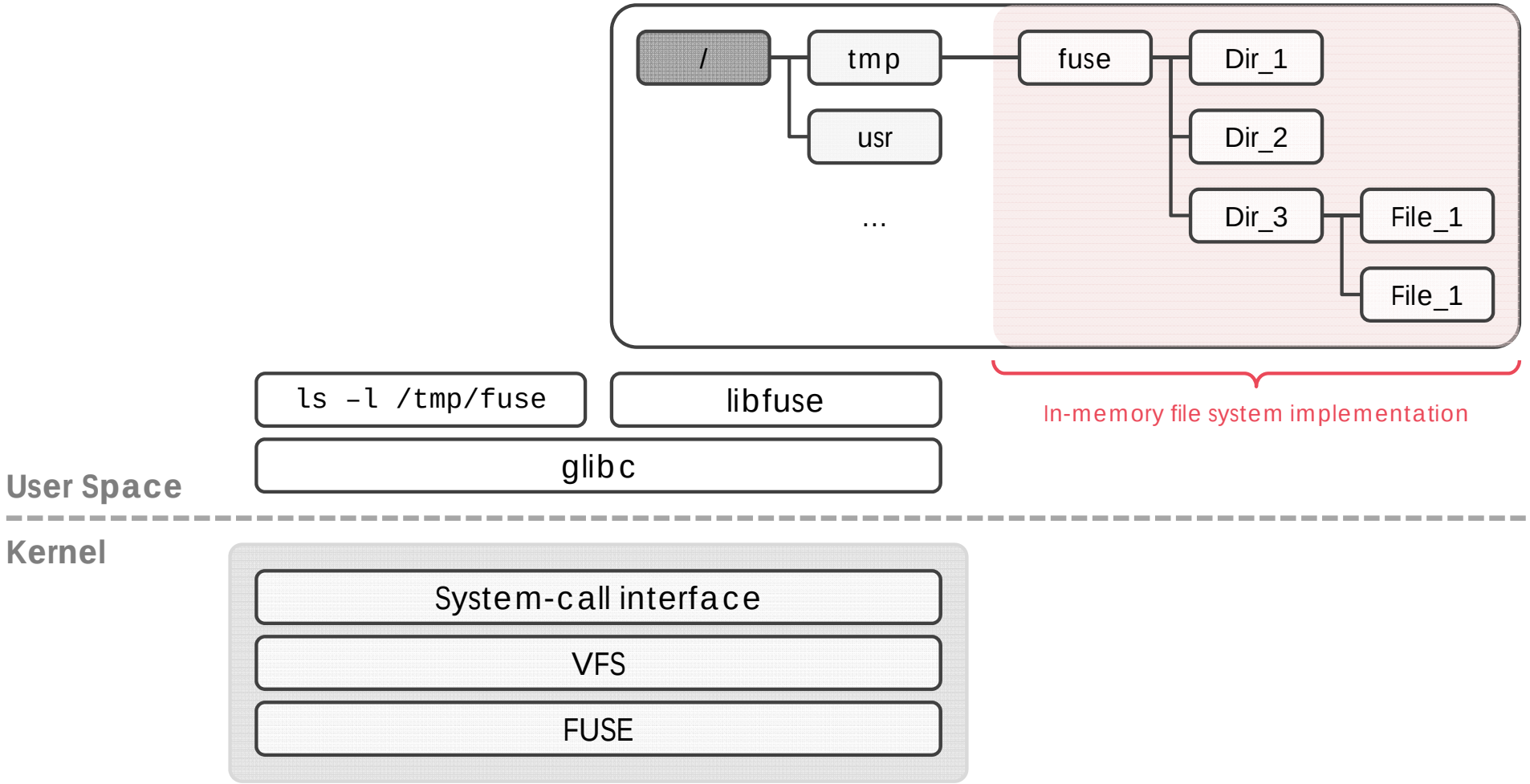
- Unmount
  - fusermount –u [mount point]

```
fusermount –u ./mnt
```

```
minhozx@dclab-PowerEdge-R940:~/Workspace$ mount | grep 'mnt'
/home/minhozx/Workspace/hello on /home/minhozx/Workspace/mnt type fuse.hello (rw,nosuid,nodev,relatime,user_id=1003,group_id=1003)
minhozx@dclab-PowerEdge-R940:~/Workspace$ cd mnt/
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$ ls -al
total 4
drwxr-xr-x 2 root    root       0  1월  1  1970 .
drwxrwxr-x 4 minhozx minhozx 4096  5월 22 23:14 ..
-r--r--r-- 1 root    root      13  1월  1  1970 hello
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$ cat hello
Hello World!
minhozx@dclab-PowerEdge-R940:~/Workspace/mnt$
```

Project

- FUSE-based in-memory file system implementation
  - ex) tree structure



Project

- FUSE-based in-memory file system implementation
  - 필수 기능 구현 [기본]
    - → 파일 열기/닫기 (open/close)
    - → 파일 읽기/쓰기 (read/write)
    - → 파일 생성/삭제 (create/remove)
    - → File descriptor를 이용한 기능 구현
    - → 다단계 디렉터리 생성 가능 ex) A/B/C/D/file
    - → 상황에 맞는 에러코드(errno) 3개 이상 사용
  - 추가 기능 구현 [가산점]
    - → 파일의 r/w/x 권한 변경
    - → 파일 이동/복사 (move/copy)
    - → 파일 링크 (link) [softlink/hardlink 구분]
    - → 기타 각자 원하는 추가 기능을 구현 가능
  - 주의사항
    - → 유저 터미널로의 직접 접근 금지 ex) printf(“error = … ”)
    - → Error를 나타낼 때 ‘return –error code’ 사용
    - → 필수 및 추가 기능에 대한 설명은 보고서에 ‘반드시’ 명시할 것

Project

- 수행방법
  - 2인 1팀으로 팀 편성
  - 팀 단위 과제 수행 및 결과물 제출
- 제출물
  - 1차 - 제안서

    → FUSE 조사/분석서

    → In-memory File System 설계서
    (자료구조 및 파일 데이터 관리 방식, 필요한 함수에 대한 설명 등)
  - 2차 - 결과보고서

    → 구현 기능 설명서

    → 실행 결과 화면 설명서 (스크린샷 첨부)

    → 소스 코드

Project

- 제출기한
  - 제안서 제출

    → 2019.06.04(화) 오전 11:00 까지

    → Hardcopy로 출력하여 27315호(분산컴퓨팅연구실)에 제출

  - 결과보고서 제출

    → 2017.06.13(목) 오전 11:00 까지

    → 결과보고서와 소스 코드를 압축하여 iCampus 과제란에 제출

References

- FUSE Official Webpage, https://github.com/libfuse/libfuse
- FUSE Document – fuse_operations list, http://fuse.sourceforge.net/doxygen/structfuse__operations.html
- https://www.cs.hmc.edu/~geoff/classes/hmc.cs135.201001/homework/fuse/fuse_doc.html
- M. Abd-El-Malek, M. Wachs, J. Cipar Gregory, R. Ganger, G. A. Gibson1, M. K. Reiter, File system virtual appliances: Third-party file system implementations without the pain, Carnegie Mellon University Panasas, Inc., 2008