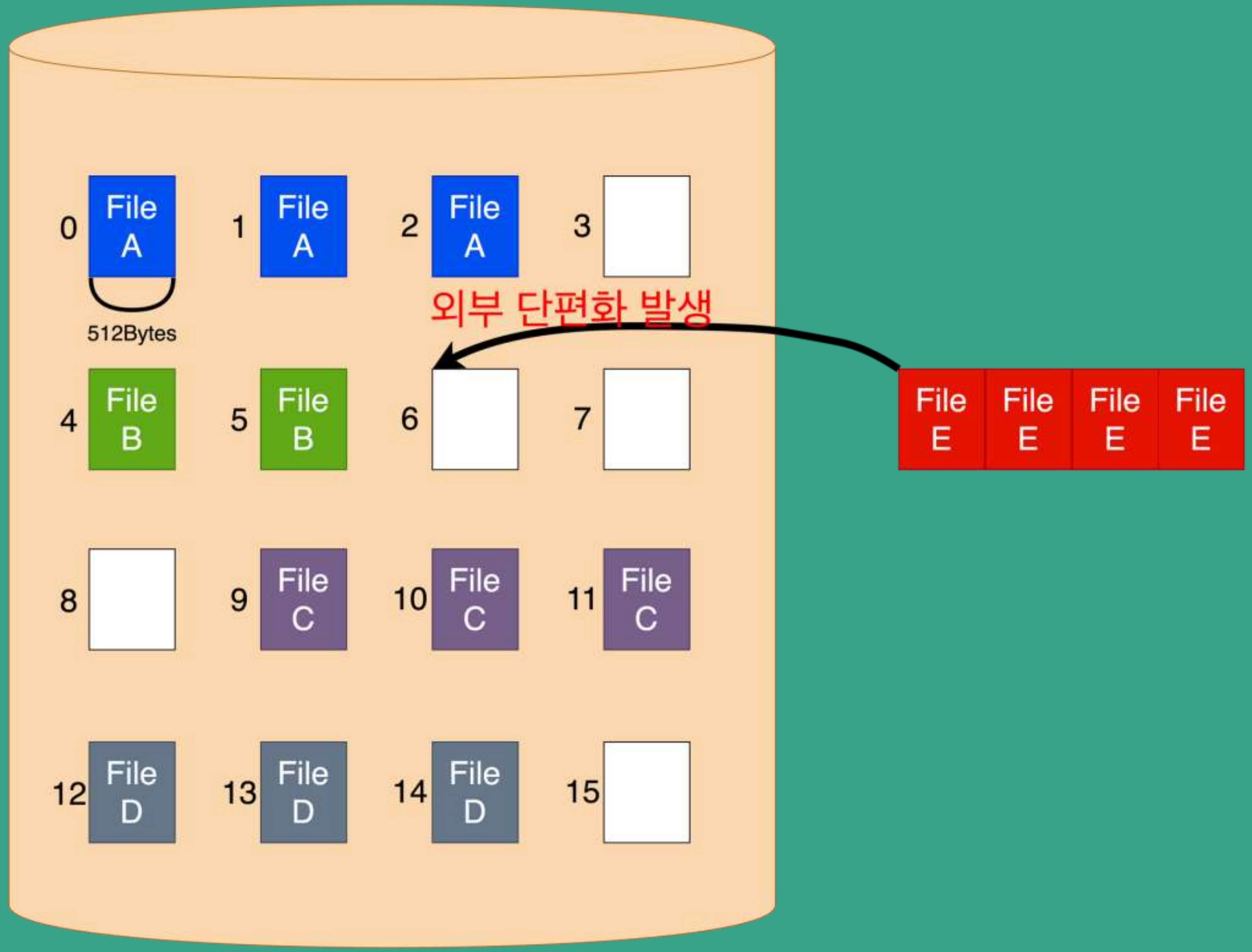
WEEK10 WIL

6조 정재혁 유흥국 남청우

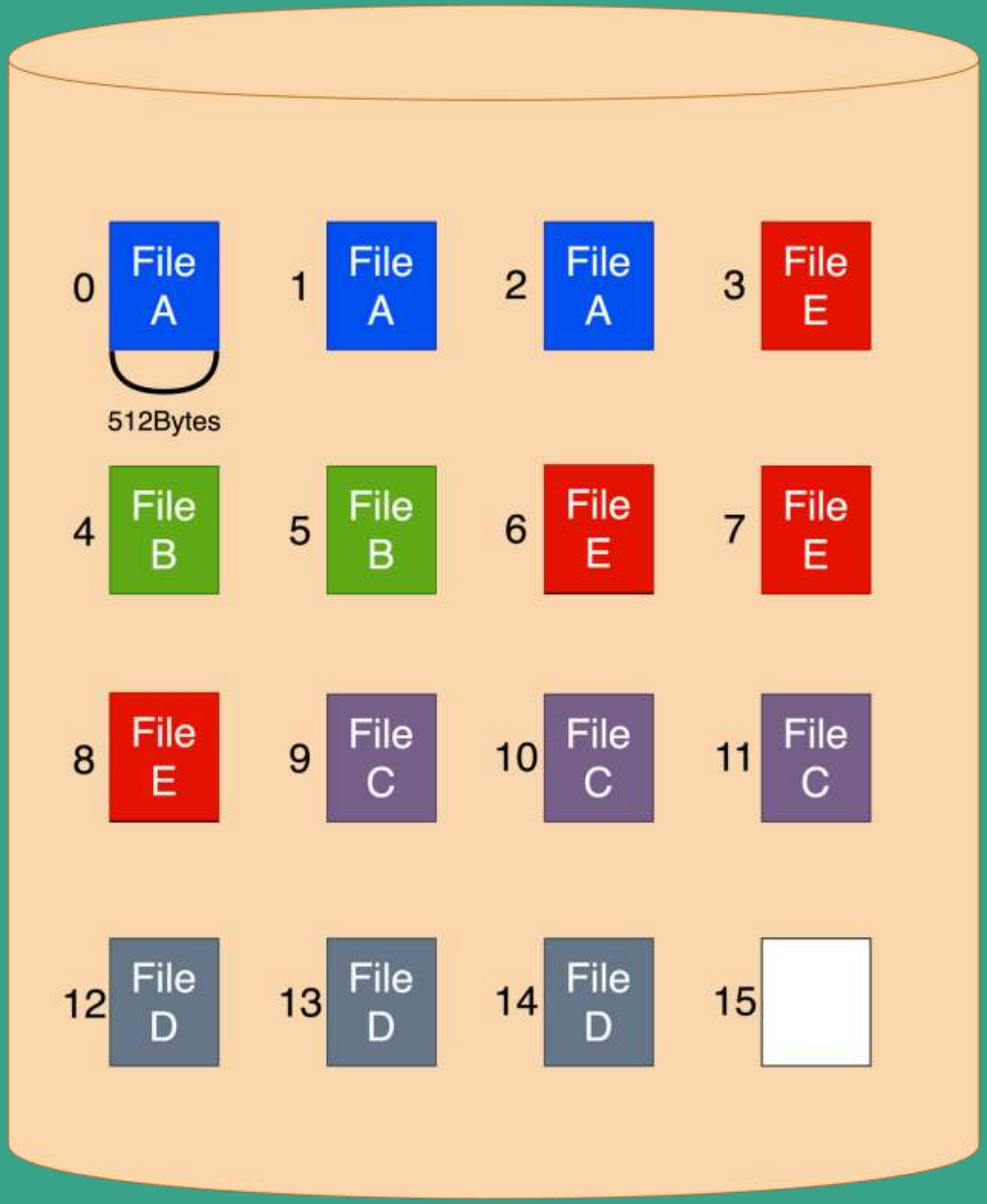
PROJECT4 FILE

SYSTEMS

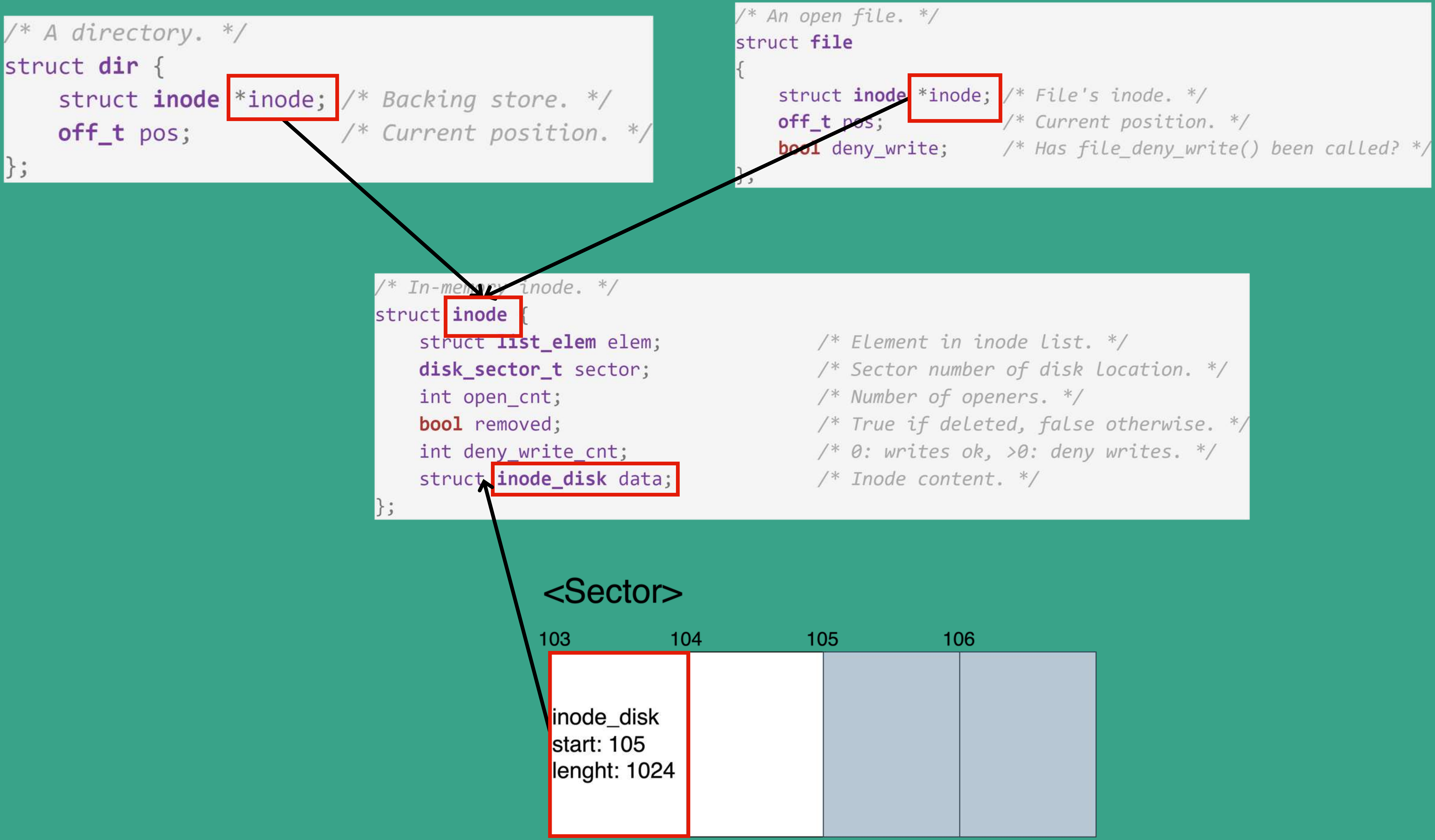
✓ 기존 PintOS


0
File A
1
File A
2
File A
3
512Bytes
외부 단편화 발생
4
File B
5
File B
6
7
File E
File E
File E
File E
8
9
File C
10
File C
11
File C
12
File D
13
File D
14
File D
15

✓ FAT



idx	val
0	1
1	2
2	EOF
3	6
4	5
5	EOF
6	7
7	8
8	EOF
9	10
10	11
11	EOF
12	13
13	14
14	EOF
15	

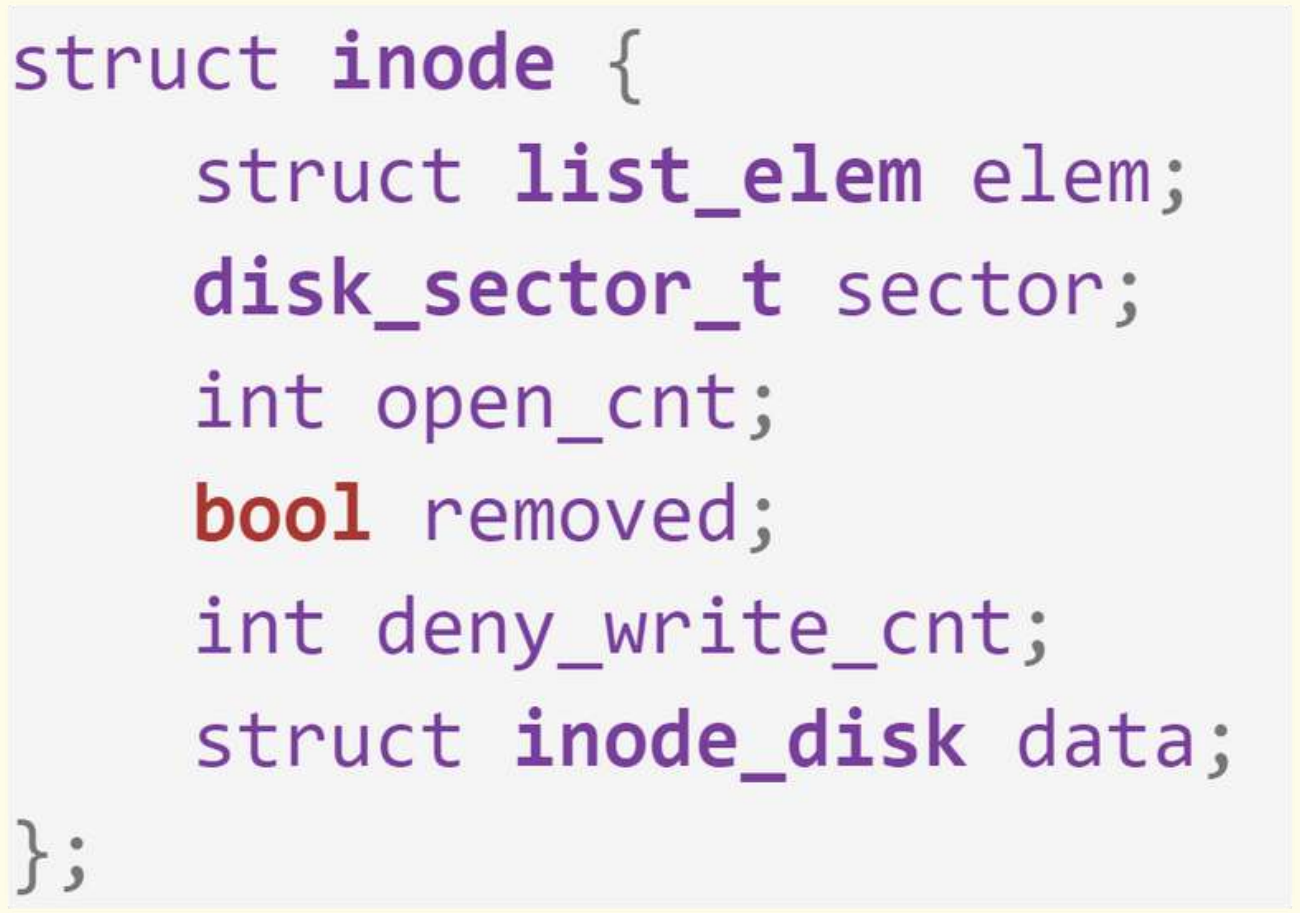


```c
/* A directory. */
struct dir {
    struct inode *inode;    /* Backing store. */
    off_t pos;              /* Current position. */
};
```

```c
/* An open file. */
struct file
{
    struct inode *inode;    /* File's inode. */
    off_t pos;              /* Current position. */
    bool deny_write;        /* Has file_deny_write() been called? */
};
```

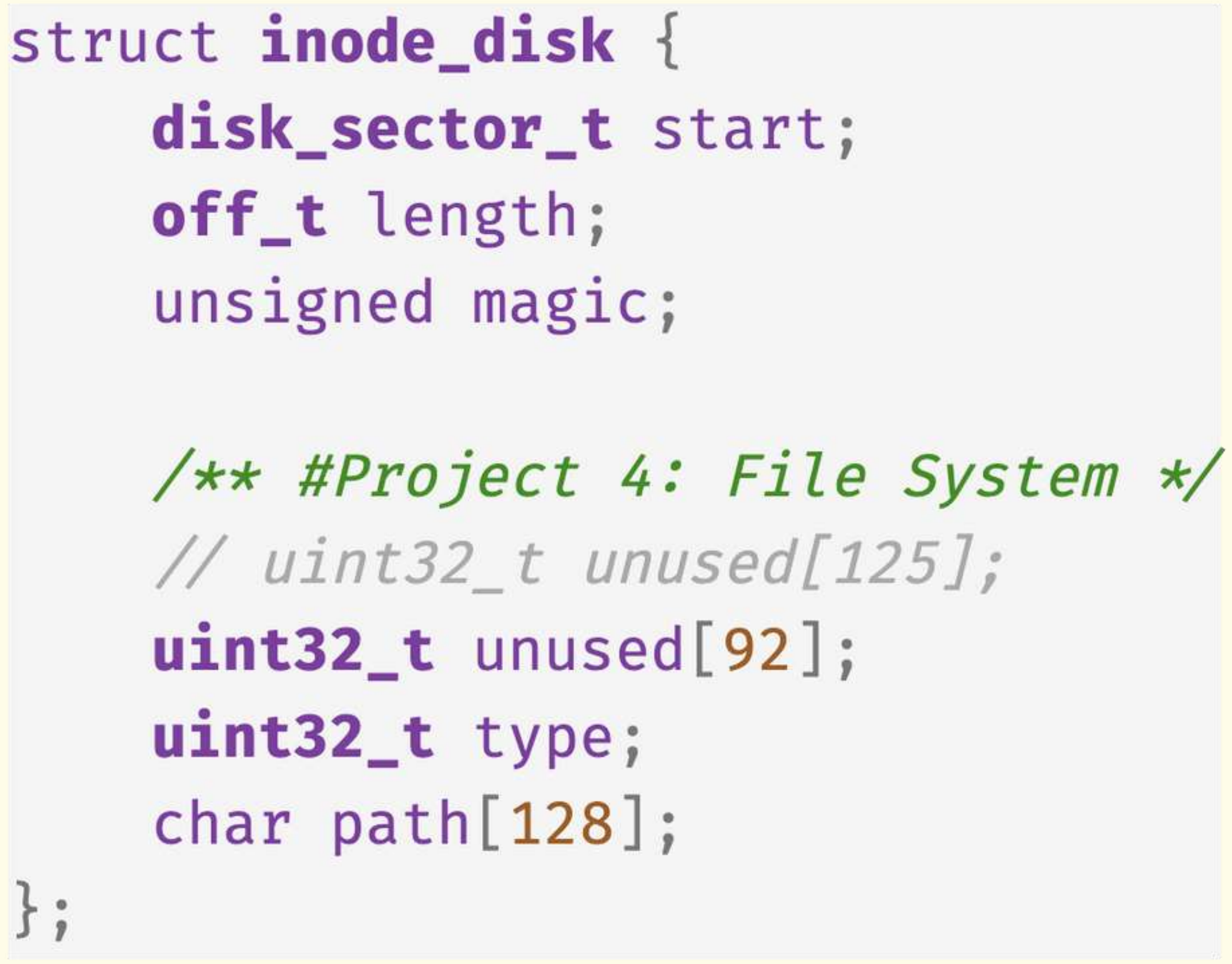
```c
/* In-memory inode. */
struct inode {
    struct list_elem elem;              /* Element in inode list. */
    disk_sector_t sector;               /* Sector number of disk location. */
    int open_cnt;                       /* Number of openers. */
    bool removed;                       /* True if deleted, false otherwise. */
    int deny_write_cnt;                 /* 0: writes ok, >0: deny writes. */
    struct inode_disk data;             /* Inode content. */
};
```

✓ Inode



```c
struct inode {
    struct list_elem elem;
    disk_sector_t sector;
    int open_cnt;
    bool removed;
    int deny_write_cnt;
    struct inode_disk data;
};
```

✓ Inode_disk



```c
struct inode_disk {
    disk_sector_t start;
    off_t length;
    unsigned magic;

    /** #Project 4: File System */
    // uint32_t unused[125];
    uint32_t unused[92];
    uint32_t type;
    char path[128];
};
```

✓ mkdir

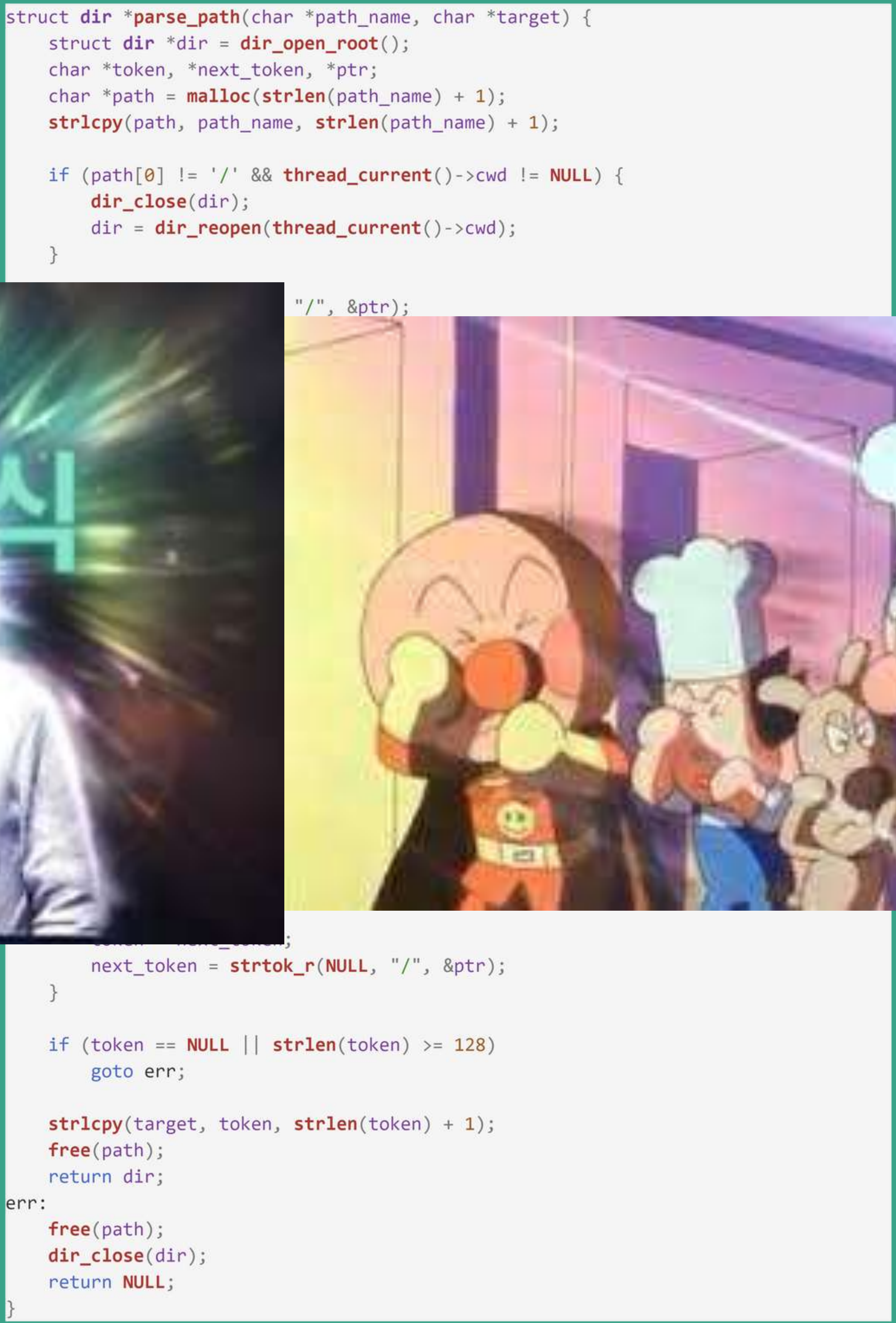
```c
bool filesys_mkdir(const char *dir_name) {
    cluster_t inode_cluster = fat_create_chain(0);
    disk_sector_t inode_sector = cluster_to_sector(inode_cluster);
    char target[128];

    if (strlen(dir_name) == 0)
        return false;

    struct dir *dir_path = parse_path(dir_name, target);
    if (dir_path == NULL)
        return false;

    struct dir *dir = dir_reopen(dir_path);

    // 할당 받은 cluster에 inode를 만들고 directory 추가
    bool success = (dir != NULL && inode_create(inode_sector, 0, DIR_TYPE)
                    && dir_add(dir, target, inode_sector));

    if (!success && inode_cluster != 0)
        fat_remove_chain(inode_cluster, 0);

    if (success) {  // directory에 .과 .. 추가
        struct inode *inode = NULL;
        dir_lookup(dir, target, &inode);
        struct dir *new_dir = dir_open(inode);

        if (!dir_add(new_dir, ".", inode_sector))
            success = false;
        if (!dir_add(new_dir, "..", inode_get_inumber(dir_get_inode(dir))))
            success = false;

        dir_close(new_dir);
    }

    dir_close(dir);

    return success;
}
```

✓ chdir

```c
bool filesys_chdir(const char *dir_name) {
    struct inode *inode = NULL;
    char target[128];
    target[0] = '\0';
    struct dir *dir = parse_path(dir_name, target);

    if (!dir_lookup(dir, target, &inode))
        return false;

    if (inode_get_type(inode) == 0 || inode_is_removed(inode))
        return false;

    dir = dir_open(inode);

    thread_current()->cwd = dir;

    return true;
}
```

✓ cwd

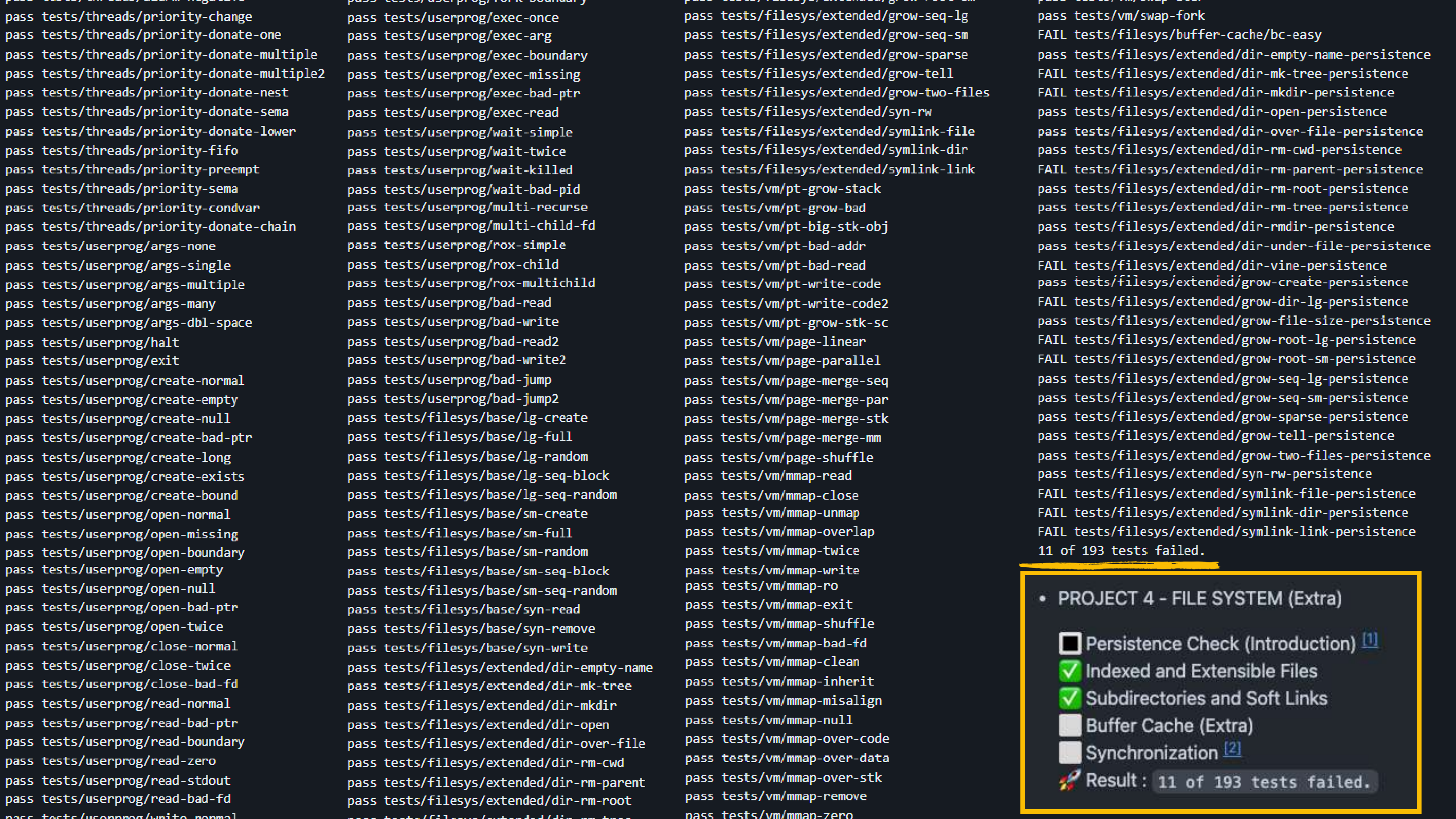
```c
    /** Project 4: Filesys - File System */
    struct dir *cwd; // Current Working Directory
```



```
struct dir *parse_path(char *path_name, char *target) {
    struct dir *dir = dir_open_root();
    char *token, *next_token, *ptr;
    char *path = malloc(strlen(path_name) + 1);
    strlcpy(path, path_name, strlen(path_name) + 1);

    if (path[0] != '/' && thread_current()->cwd != NULL) {
        dir_close(dir);
        dir = dir_reopen(thread_current()->cwd);
    }

                      "/", &ptr);

        next_token = strtok_r(NULL, "/", &ptr);
    }

    if (token == NULL || strlen(token) >= 128)
        goto err;

    strlcpy(target, token, strlen(token) + 1);
    free(path);
    return dir;
err:
    free(path);
    dir_close(dir);
    return NULL;
}
```

✓ Soft Link

```
bool filesys_symlink(const char *target, const char *linkpath) {
    cluster_t inode_cluster = fat_create_chain(0);
    disk_sector_t inode_sector = cluster_to_sector(inode_cluster);

    struct inode *target_inode = NULL;
    struct inode *inode = NULL;
    bool success;

    char link_name[128];
    link_name[0] = '\0';

    struct dir *link_dir = parse_path(linkpath, link_name);

    if (strcmp(link_name, "") == 0)
        return false;

    if (link_dir == NULL || inode_is_removed(dir_get_inode(link_dir)))
        return false;

    success = (link_dir != NULL && inode_create(inode_sector, 0, LINK_TYPE) && dir_add(link_dir, link_name, inode_sector));

    if (!success && inode_sector != 0) {
        fat_remove_chain(inode_cluster, 1);
        return success;
    }

    dir_lookup(link_dir, link_name, &inode);

    inode_set_linkpath(inode, target);

    return success;
}
```



```
pass tests/threads/alarm-negative
pass tests/threads/priority-change
pass tests/threads/priority-donate-one
pass tests/threads/priority-donate-multiple
pass tests/threads/priority-donate-multiple2
pass tests/threads/priority-donate-nest
pass tests/threads/priority-donate-sema
pass tests/threads/priority-donate-lower
pass tests/threads/priority-fifo
pass tests/threads/priority-preempt
pass tests/threads/priority-sema
pass tests/threads/priority-condvar
pass tests/threads/priority-donate-chain
pass tests/userprog/args-none
pass tests/userprog/args-single
pass tests/userprog/args-multiple
pass tests/userprog/args-many
pass tests/userprog/args-dbl-space
pass tests/userprog/halt
pass tests/userprog/exit
pass tests/userprog/create-normal
pass tests/userprog/create-empty
pass tests/userprog/create-null
pass tests/userprog/create-bad-ptr
pass tests/userprog/create-long
pass tests/userprog/create-exists
pass tests/userprog/create-bound
pass tests/userprog/open-normal
pass tests/userprog/open-missing
pass tests/userprog/open-boundary
pass tests/userprog/open-empty
pass tests/userprog/open-null
pass tests/userprog/open-bad-ptr
pass tests/userprog/open-twice
pass tests/userprog/close-normal
pass tests/userprog/close-twice
pass tests/userprog/close-bad-fd
pass tests/userprog/read-normal
pass tests/userprog/read-bad-ptr
pass tests/userprog/read-boundary
pass tests/userprog/read-zero
pass tests/userprog/read-stdout
pass tests/userprog/read-bad-fd
pass tests/userprog/write-normal
```

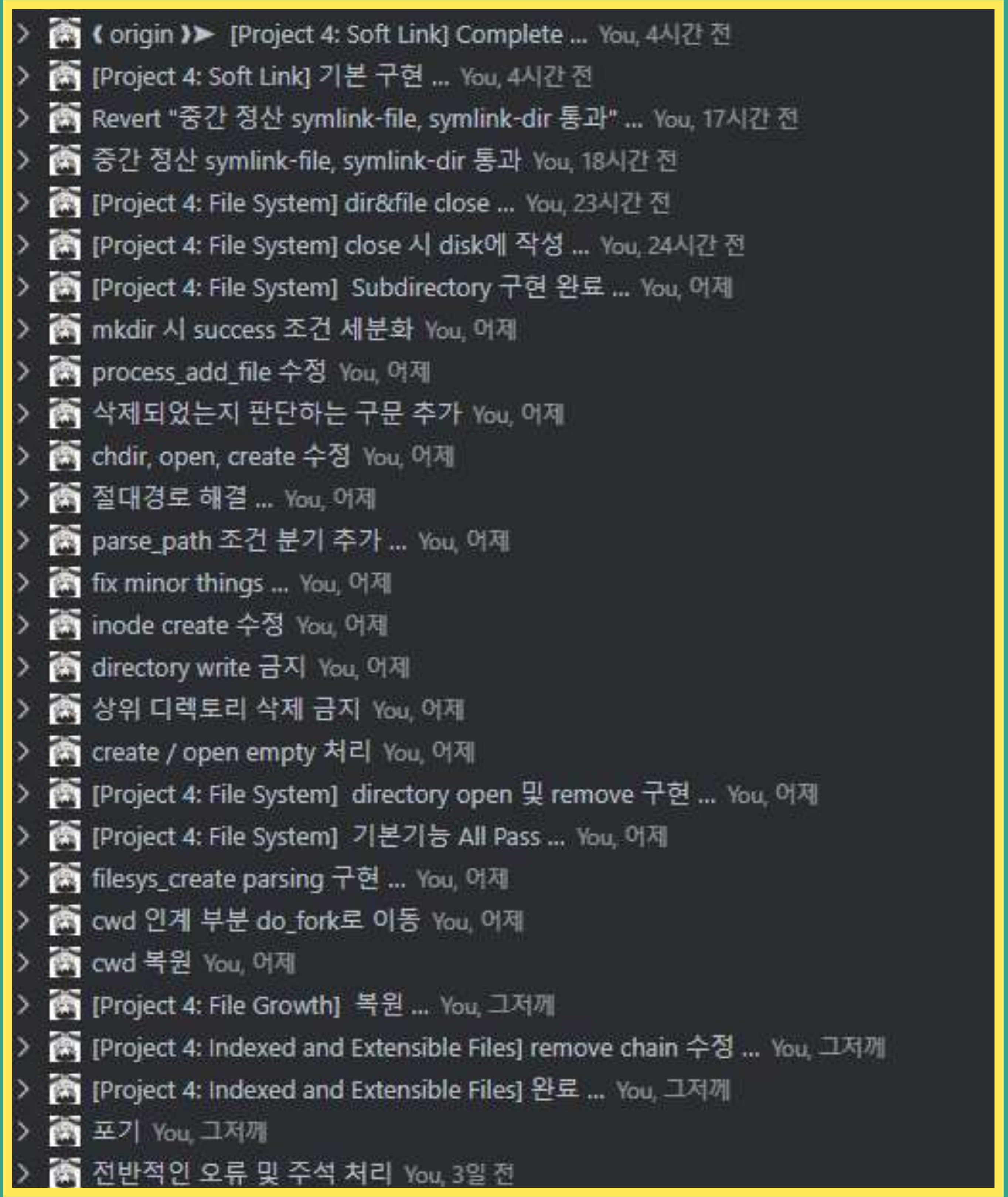
```
pass tests/userprog/fork-boundary
pass tests/userprog/exec-once
pass tests/userprog/exec-arg
pass tests/userprog/exec-boundary
pass tests/userprog/exec-missing
pass tests/userprog/exec-bad-ptr
pass tests/userprog/exec-read
pass tests/userprog/wait-simple
pass tests/userprog/wait-twice
pass tests/userprog/wait-killed
pass tests/userprog/wait-bad-pid
pass tests/userprog/multi-recurse
pass tests/userprog/multi-child-fd
pass tests/userprog/rox-simple
pass tests/userprog/rox-child
pass tests/userprog/rox-multichild
pass tests/userprog/bad-read
pass tests/userprog/bad-write
pass tests/userprog/bad-read2
pass tests/userprog/bad-write2
pass tests/userprog/bad-jump
pass tests/userprog/bad-jump2
pass tests/filesys/base/lg-create
pass tests/filesys/base/lg-full
pass tests/filesys/base/lg-random
pass tests/filesys/base/lg-seq-block
pass tests/filesys/base/lg-seq-random
pass tests/filesys/base/sm-create
pass tests/filesys/base/sm-full
pass tests/filesys/base/sm-random
pass tests/filesys/base/sm-seq-block
pass tests/filesys/base/sm-seq-random
pass tests/filesys/base/syn-read
pass tests/filesys/base/syn-remove
pass tests/filesys/base/syn-write
pass tests/filesys/extended/dir-empty-name
pass tests/filesys/extended/dir-mk-tree
pass tests/filesys/extended/dir-mkdir
pass tests/filesys/extended/dir-open
pass tests/filesys/extended/dir-over-file
pass tests/filesys/extended/dir-rm-cwd
pass tests/filesys/extended/dir-rm-parent
pass tests/filesys/extended/dir-rm-root
pass tests/filesys/extended/dir-rm-tree
```

```
pass tests/filesys/extended/grow-root-sm
pass tests/filesys/extended/grow-seq-lg
pass tests/filesys/extended/grow-seq-sm
pass tests/filesys/extended/grow-sparse
pass tests/filesys/extended/grow-tell
pass tests/filesys/extended/grow-two-files
pass tests/filesys/extended/syn-rw
pass tests/filesys/extended/symlink-file
pass tests/filesys/extended/symlink-dir
pass tests/filesys/extended/symlink-link
pass tests/vm/pt-grow-stack
pass tests/vm/pt-grow-bad
pass tests/vm/pt-big-stk-obj
pass tests/vm/pt-bad-addr
pass tests/vm/pt-bad-read
pass tests/vm/pt-write-code
pass tests/vm/pt-write-code2
pass tests/vm/pt-grow-stk-sc
pass tests/vm/page-linear
pass tests/vm/page-parallel
pass tests/vm/page-merge-seq
pass tests/vm/page-merge-par
pass tests/vm/page-merge-stk
pass tests/vm/page-merge-mm
pass tests/vm/page-shuffle
pass tests/vm/mmap-read
pass tests/vm/mmap-close
pass tests/vm/mmap-unmap
pass tests/vm/mmap-overlap
pass tests/vm/mmap-twice
pass tests/vm/mmap-write
pass tests/vm/mmap-ro
pass tests/vm/mmap-exit
pass tests/vm/mmap-shuffle
pass tests/vm/mmap-bad-fd
pass tests/vm/mmap-clean
pass tests/vm/mmap-inherit
pass tests/vm/mmap-misalign
pass tests/vm/mmap-null
pass tests/vm/mmap-over-code
pass tests/vm/mmap-over-data
pass tests/vm/mmap-over-stk
pass tests/vm/mmap-remove
pass tests/vm/mmap-zero
```

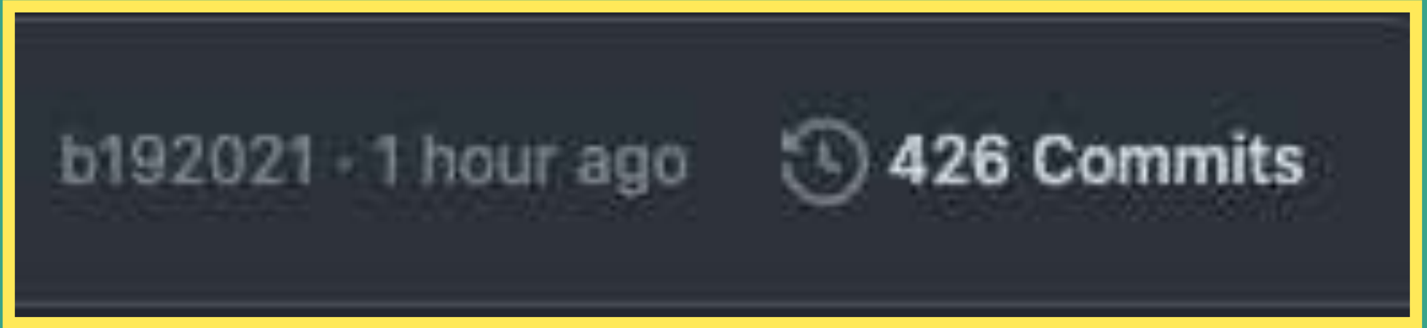
```
pass tests/vm/swap-iter
pass tests/vm/swap-fork
FAIL tests/filesys/buffer-cache/bc-easy
pass tests/filesys/extended/dir-empty-name-persistence
FAIL tests/filesys/extended/dir-mk-tree-persistence
FAIL tests/filesys/extended/dir-mkdir-persistence
pass tests/filesys/extended/dir-open-persistence
pass tests/filesys/extended/dir-over-file-persistence
pass tests/filesys/extended/dir-rm-cwd-persistence
FAIL tests/filesys/extended/dir-rm-parent-persistence
pass tests/filesys/extended/dir-rm-root-persistence
pass tests/filesys/extended/dir-rm-tree-persistence
pass tests/filesys/extended/dir-rmdir-persistence
pass tests/filesys/extended/dir-under-file-persistence
FAIL tests/filesys/extended/dir-vine-persistence
pass tests/filesys/extended/grow-create-persistence
FAIL tests/filesys/extended/grow-dir-lg-persistence
pass tests/filesys/extended/grow-file-size-persistence
FAIL tests/filesys/extended/grow-root-lg-persistence
FAIL tests/filesys/extended/grow-root-sm-persistence
pass tests/filesys/extended/grow-seq-lg-persistence
pass tests/filesys/extended/grow-seq-sm-persistence
pass tests/filesys/extended/grow-sparse-persistence
pass tests/filesys/extended/grow-tell-persistence
pass tests/filesys/extended/grow-two-files-persistence
pass tests/filesys/extended/syn-rw-persistence
FAIL tests/filesys/extended/symlink-file-persistence
FAIL tests/filesys/extended/symlink-dir-persistence
FAIL tests/filesys/extended/symlink-link-persistence
11 of 193 tests failed.
```

- PROJECT 4 - FILE SYSTEM (Extra)
  - ⬛ Persistence Check (Introduction) [1]
  - ✅ Indexed and Extensible Files
  - ✅ Subdirectories and Soft Links
  - ⬜ Buffer Cache (Extra)
  - ⬜ Synchronization [2]
  - 🚀 Result : `11 of 193 tests failed.`

✓ 느낀점



```
> ⟨ origin ⟩➤ [Project 4: Soft Link] Complete ... You, 4시간 전
> [Project 4: Soft Link] 기본 구현 ... You, 4시간 전
> Revert "중간 정산 symlink-file, symlink-dir 통과" ... You, 17시간 전
> 중간 정산 symlink-file, symlink-dir 통과 You, 18시간 전
> [Project 4: File System] dir&file close ... You, 23시간 전
> [Project 4: File System] close 시 disk에 작성 ... You, 24시간 전
> [Project 4: File System]  Subdirectory 구현 완료 ... You, 어제
> mkdir 시 success 조건 세분화 You, 어제
> process_add_file 수정 You, 어제
> 삭제되었는지 판단하는 구문 추가 You, 어제
> chdir, open, create 수정 You, 어제
> 절대경로 해결 ... You, 어제
> parse_path 조건 분기 추가 ... You, 어제
> fix minor things ... You, 어제
> inode create 수정 You, 어제
> directory write 금지 You, 어제
> 상위 디렉토리 삭제 금지 You, 어제
> create / open empty 처리 You, 어제
> [Project 4: File System]  directory open 및 remove 구현 ... You, 어제
> [Project 4: File System]  기본기능 All Pass ... You, 어제
> filesys_create parsing 구현 ... You, 어제
> cwd 인계 부분 do_fork로 이동 You, 어제
> cwd 복원 You, 어제
> [Project 4: File Growth]  복원 ... You, 그저께
> [Project 4: Indexed and Extensible Files] remove chain 수정 ... You, 그저께
> [Project 4: Indexed and Extensible Files] 완료 ... You, 그저께
> 포기 You, 그저께
> 전반적인 오류 및 주석 처리 You, 3일 전
```



b192021 · 1 hour ago 426 Commits

감사합니다