



CS 1550

Week 13

—

Project 4

Teaching Assistant

Maher Khan

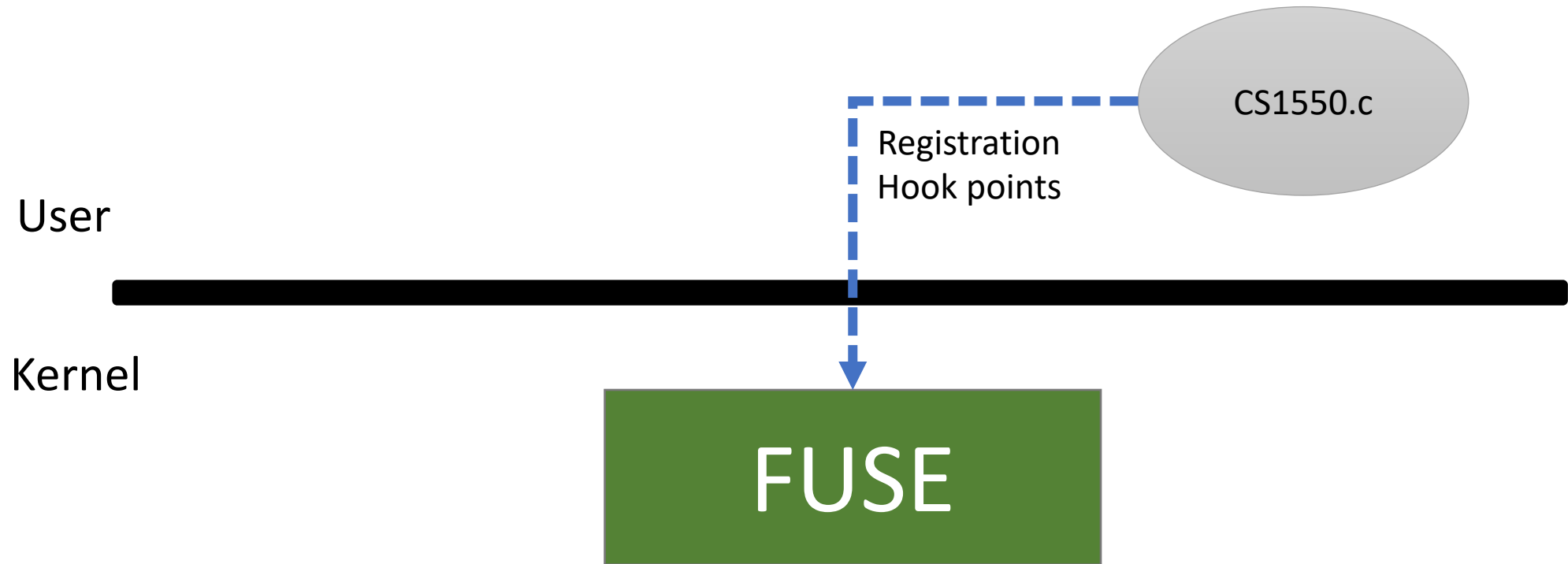
Overview

- FUSE is a **Linux kernel extension** that allows for a user space program to provide the implementations for the various file-related syscalls
- Goal: Use FUSE to create our own file system

Overview



Overview

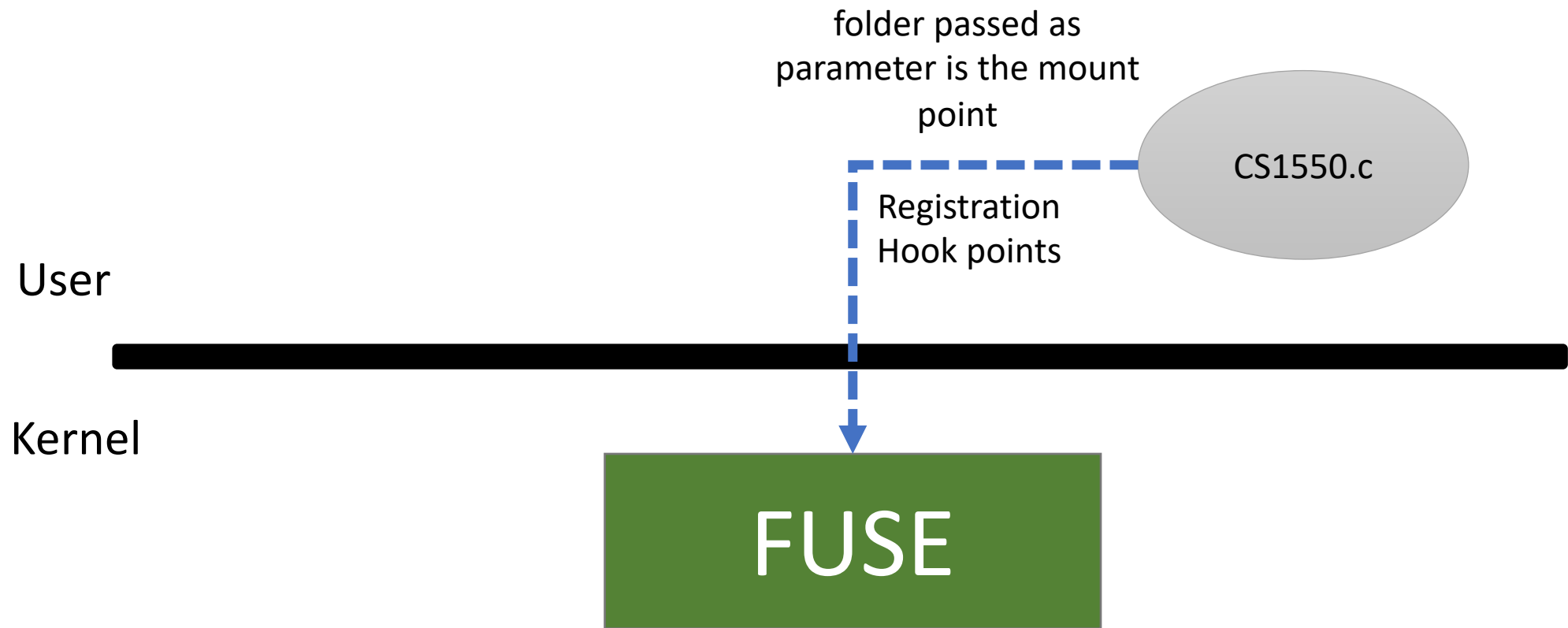


Overview

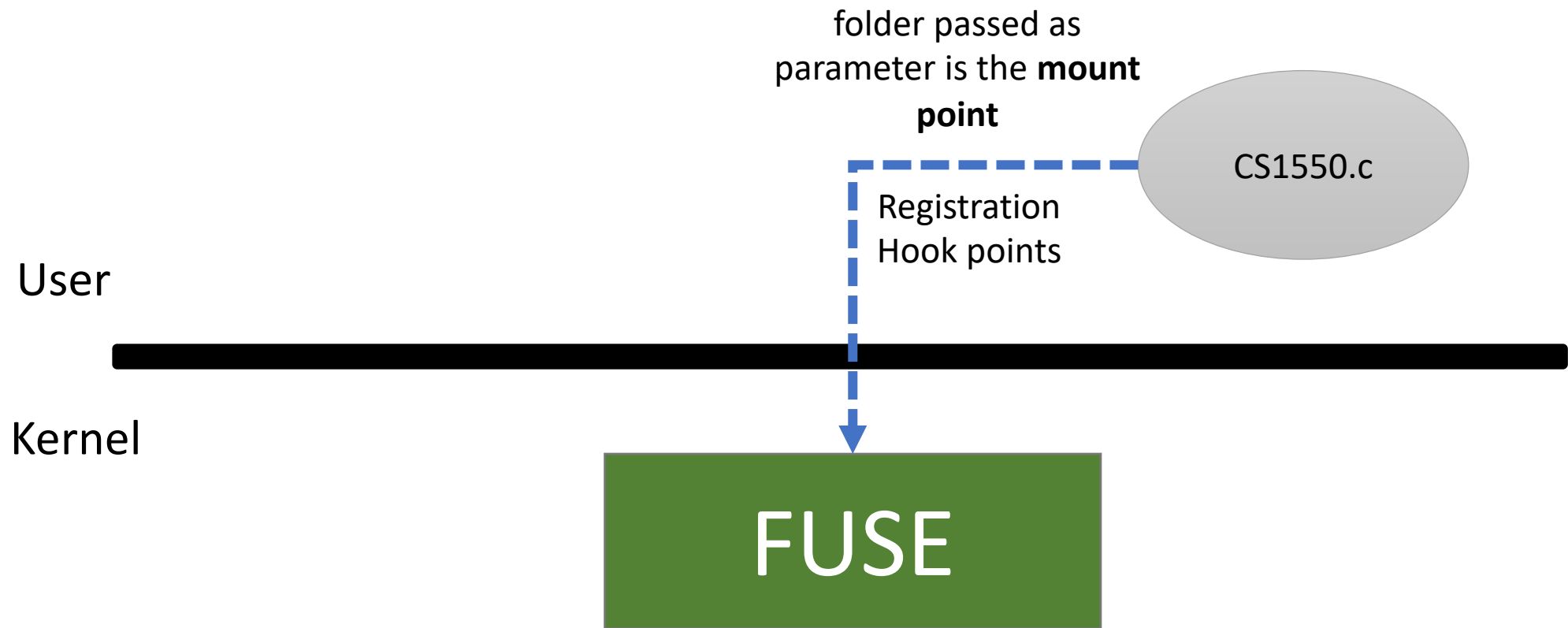
- In hello.c

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

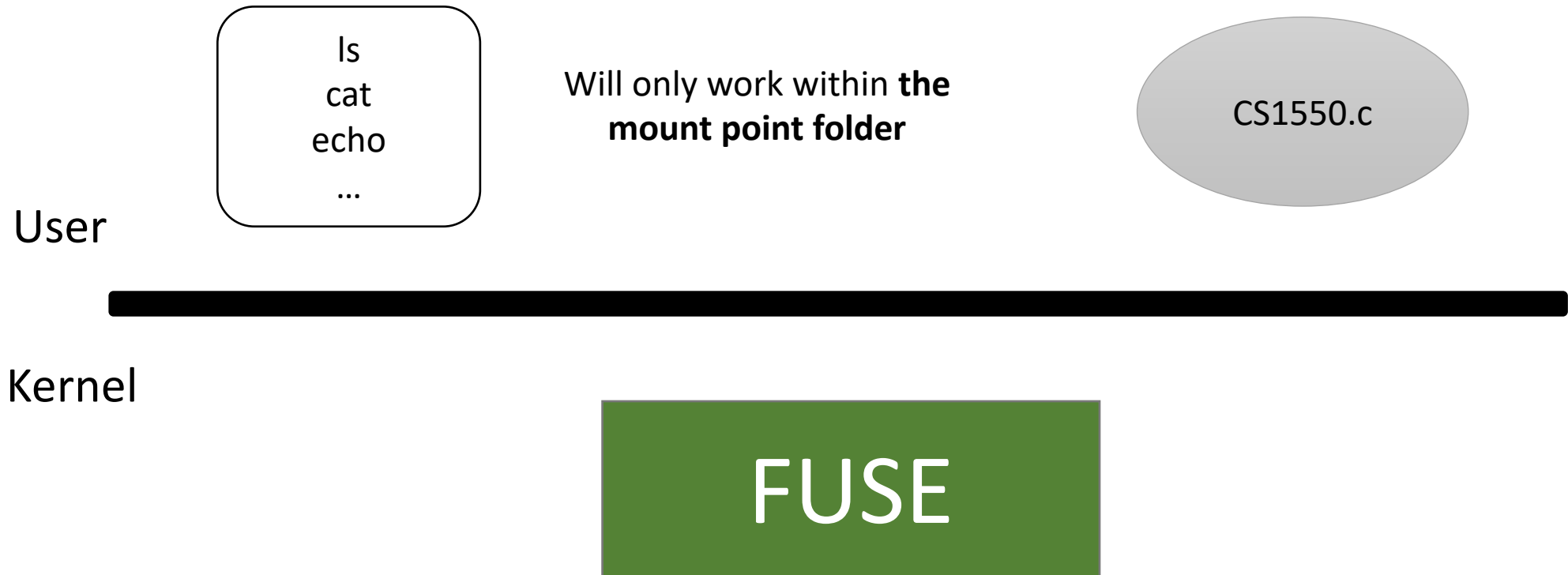
Overview



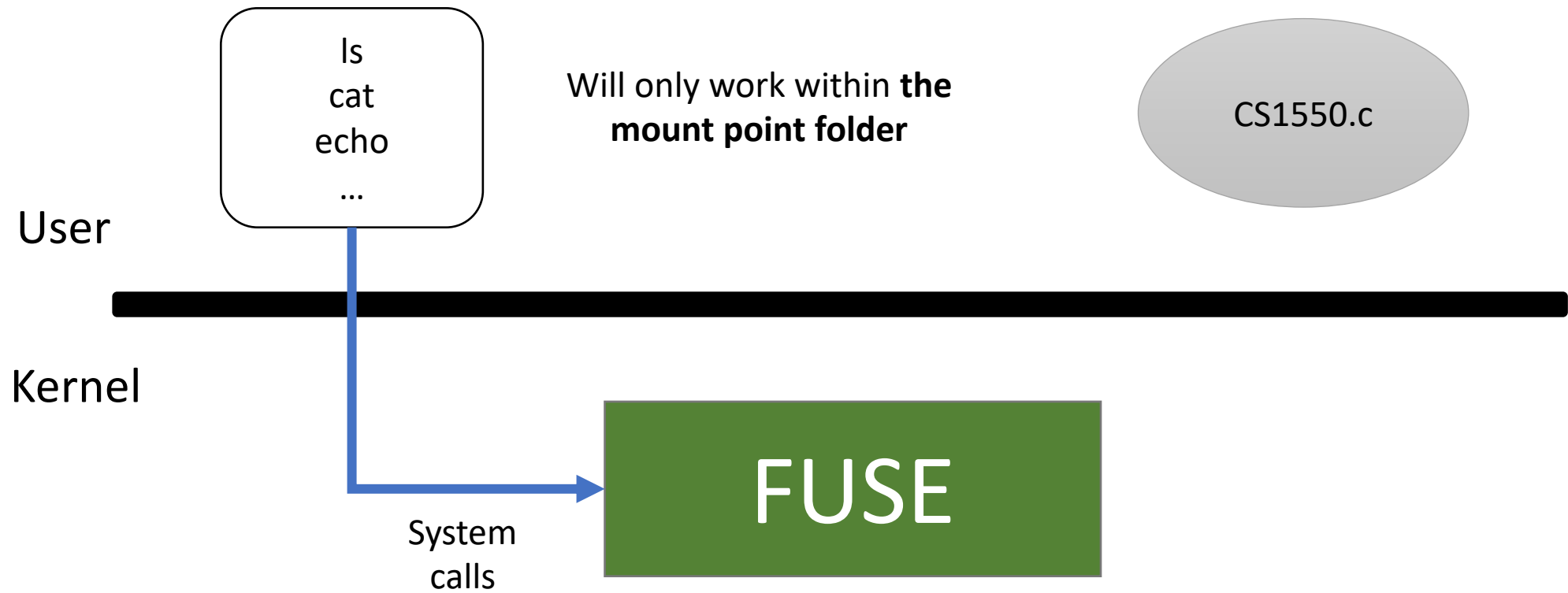
Overview



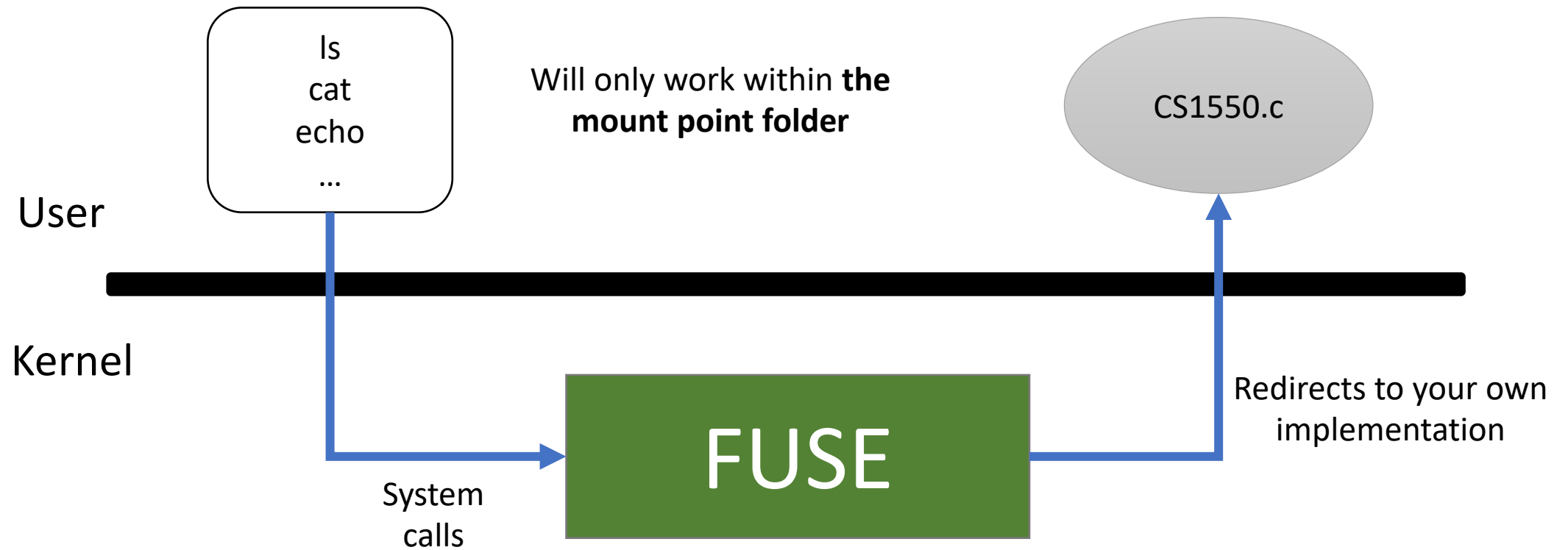
Overview



Overview



Overview



Installation of FUSE

- Kernel is installed
- Install libraries and example programs

```
cd /u/OSLab/USERNAME
```

```
cp /u/OSLab/original/fuse-2.7.0.tar.gz .
```

```
tar xvfz fuse-2.7.0.tar.gz
```

```
cd fuse-2.7.0
```

```
./configure
```

```
make
```

Installation of FUSE

- Kernel is installed
- Install libraries and example programs

```
cd /u/OSLab/USERNAME
```

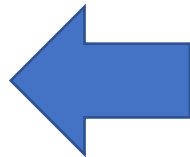
```
cp /u/OSLab/original/fuse-2.7.0.tar.gz .
```

```
tar xvfz fuse-2.7.0.tar.gz
```

```
cd fuse-2.7.0
```

```
./configure
```

```
make
```



This compiles the examples.

FUSE Example

```
cd /u/OSLab/USERNAME/
```

FUSE Example

```
cd /u/OSLab/USERNAME/  
cd fuse-2.7.0/example
```

FUSE Example

```
cd /u/OSLab/USERNAME/
```

```
cd fuse-2.7.0/example
```

```
mkdir testmount (create mount point)
```

A mount point is a location in the UNIX hierarchical file system where a new device or file system is located

FUSE Example

```
cd /u/OSLab/USERNAME/
```

```
cd fuse-2.7.0/example
```

```
mkdir testmount (create mount point)
```

A mount point is a location in the UNIX hierarchical file system where a new device or file system is located

```
ls -al testmount
```


FUSE Example

```
cd /u/OSLab/USERNAME/
```

```
cd fuse-2.7.0/example
```

```
mkdir testmount (create mount point)
```

A mount point is a location in the UNIX hierarchical file system where a new device or file system is located

```
ls -al testmount
```

```
./hello testmount
```

FUSE Example

```
cd /u/OSLab/USERNAME/
```

```
cd fuse-2.7.0/example
```

```
mkdir testmount (create mount point)
```

A mount point is a location in the UNIX hierarchical file system where a new device or file system is located

```
ls -al testmount
```

```
./hello testmount
```

```
ls -al testmount
```

FUSE Example

```
cd /u/OSLab/USERNAME/
```

```
cd fuse-2.7.0/example
```

```
mkdir testmount (create mount point)
```

A mount point is a location in the UNIX hierarchical file system where a new device or file system is located

```
ls -al testmount
```

```
./hello testmount
```

```
ls -al testmount
```

Should see . , .., hello

Solve the permission deny

1. Follow the instruction in the project description
2. Alternatively, add /u/fuse/bin into PATH env by typing the following 2 commands:

```
echo "export PATH=\"\$PATH:/u/fuse/bin\"" >> ~/.bash_profile  
source ~/.bash_profile
```

3. Alternatively, add an alias:

```
echo "alias fusermount='/u/fuse/bin/fusermount'" >> ~/.bash_profile  
source ~/.bash_profile
```

FUSE Example cont.

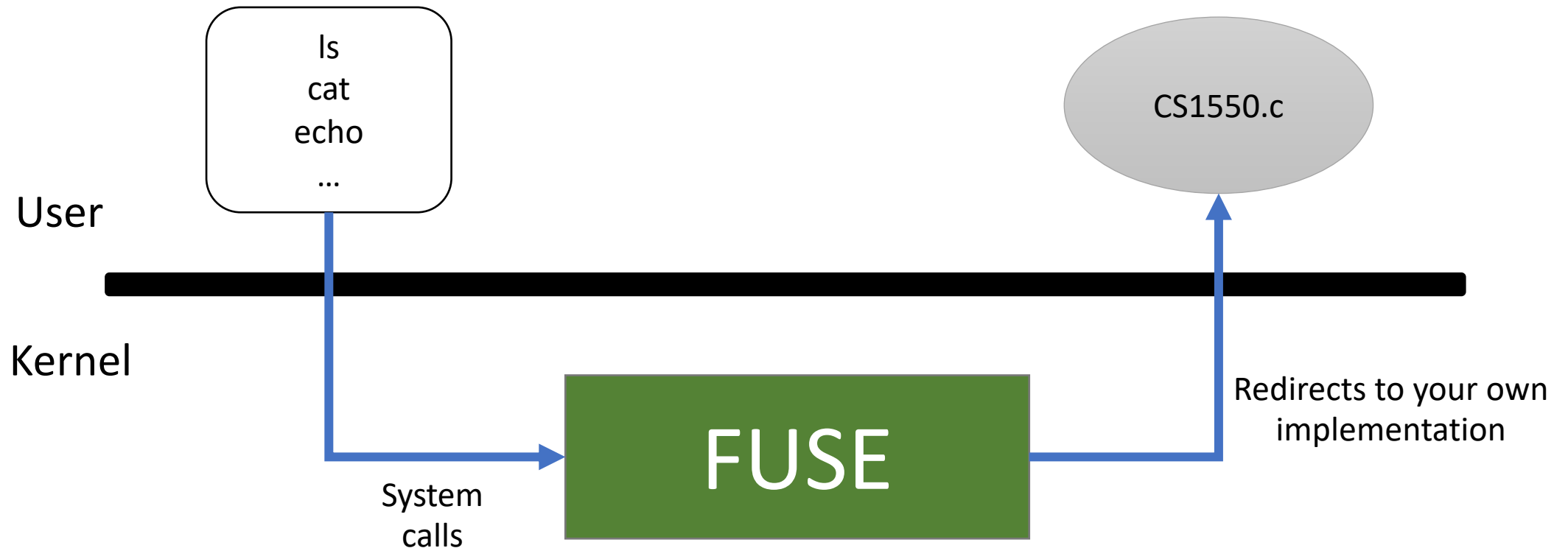
cat testmount/hello

- Hello world
- If we cat a file that doesn't really exist, how do we get meaningful output?

FUSE Example cont.

cat testmount/hello

- Hello world
- If we cat a file that doesn't really exist, how do we get meaningful output?



FUSE Example cont.

```
static int hello_read(const char *path, char *buf, size_t
size, off_t offset, struct fuse_file_info *fi)
{
    ...
}
```

FUSE Example cont.

```
static int hello_read(const char *path, char *buf, size_t
size, off_t offset, struct fuse_file_info *fi)
{
    ...
    if (offset < len) {
        ...
        memcpy(buf, hello_str + offset, size);
    } else
        size = 0;
    return size;
}
```


FUSE Example cont.

- Unmount the file system

fusermount -u testmount

What You Need To Do

- Create the **cs1550 file system** as a **FUSE application**

What You Need To Do

- Create the cs1550 file system as a FUSE application
- A code skeleton has been provided **under the FUSE zip examples** directory as cs1550.c

What You Need To Do

- Create the cs1550 file system as a FUSE application
- A code skeleton has been provided under the FUSE zip examples directory as cs1550.c
- **Automatically built** when make

What You Need To Do

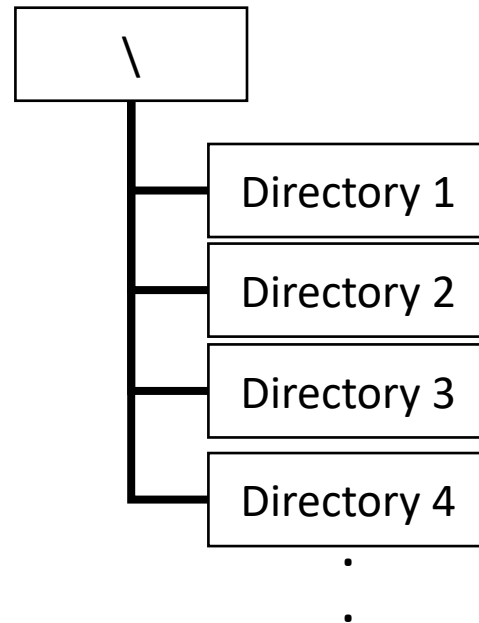
- Create the cs1550 file system as a FUSE application
- A code skeleton has been provided under the FUSE zip examples directory as cs1550.c
- Automatically built when make
- Implement **using a single file**, named **.disk 512-byte blocks**

File System

- Two-level directory system
 - The root directory “\” will only contain other subdirectories, and no regular files.

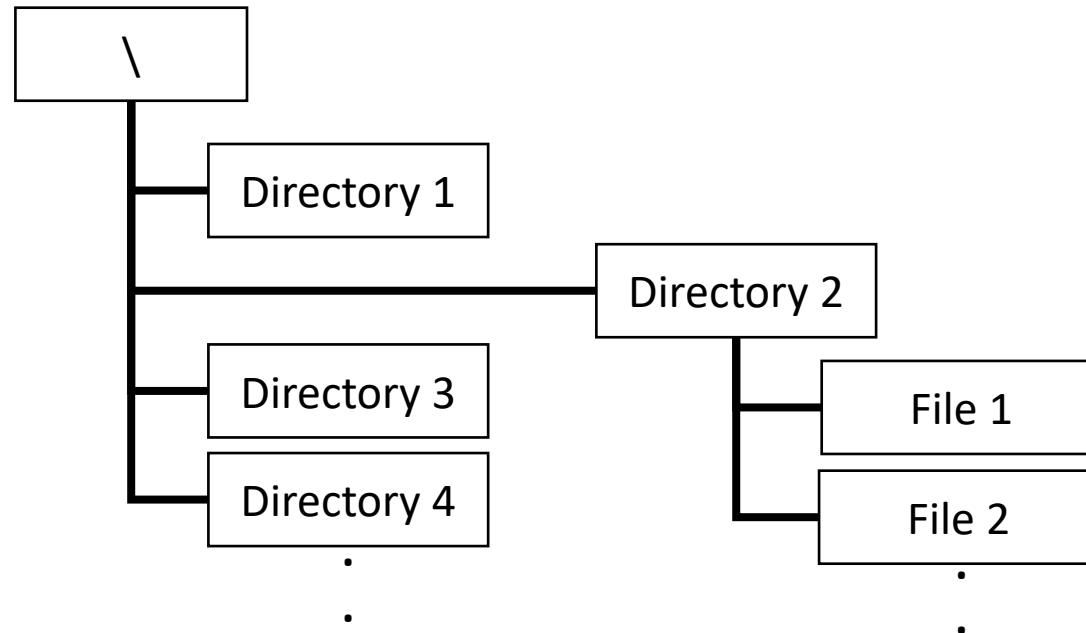
File System

- Two-level directory system
 - The root directory “\” **will only contain other subdirectories**, and no regular files.



File System

- Two-level directory system
 - The root directory “\” will only contain other subdirectories, and no regular files.
 - The subdirectories **will only contain regular files**, and no subdirectories of their own.



File System

- Two-level directory system
 - The root directory “\” will only contain other subdirectories, and no regular files.
 - The subdirectories will only contain regular files, and no subdirectories of their own.
 - All files **will be full access** with permissions to be mainly ignored.

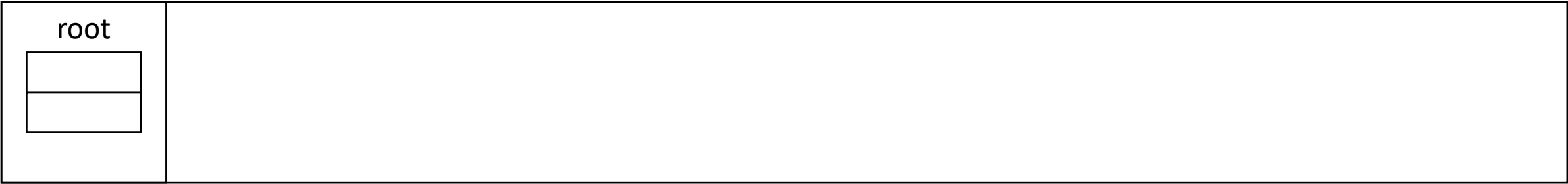
File System

- Two-level directory system
 - The root directory “\” will only contain other subdirectories, and no regular files.
 - The subdirectories will only contain regular files, and no subdirectories of their own.
 - All files will be full access with permissions to be mainly ignored.
 - Many file attributes such as creation and modification times **will not be accurately stored**.

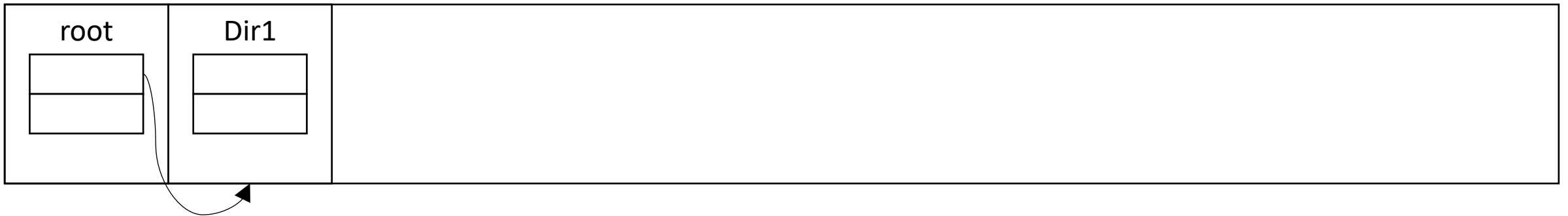
File System

- Two-level directory system
 - The root directory “\” will only contain other subdirectories, and no regular files.
 - The subdirectories will only contain regular files, and no subdirectories of their own.
 - All files will be full access with permissions to be mainly ignored.
 - Many file attributes such as creation and modification times will not be accurately stored.
 - **Files cannot be truncated.**

Structure



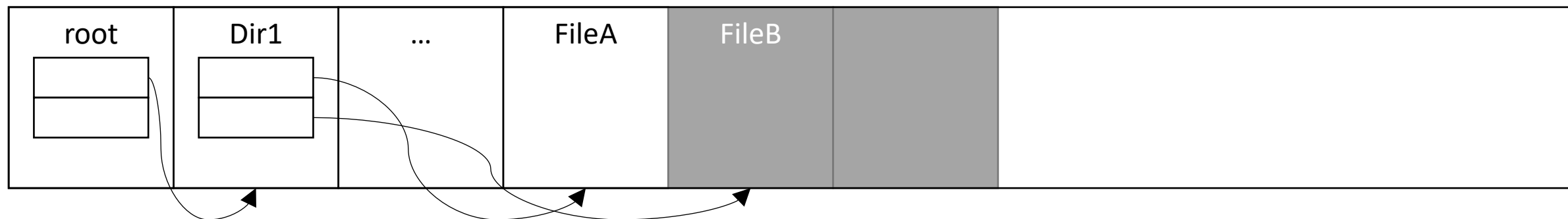
Structure



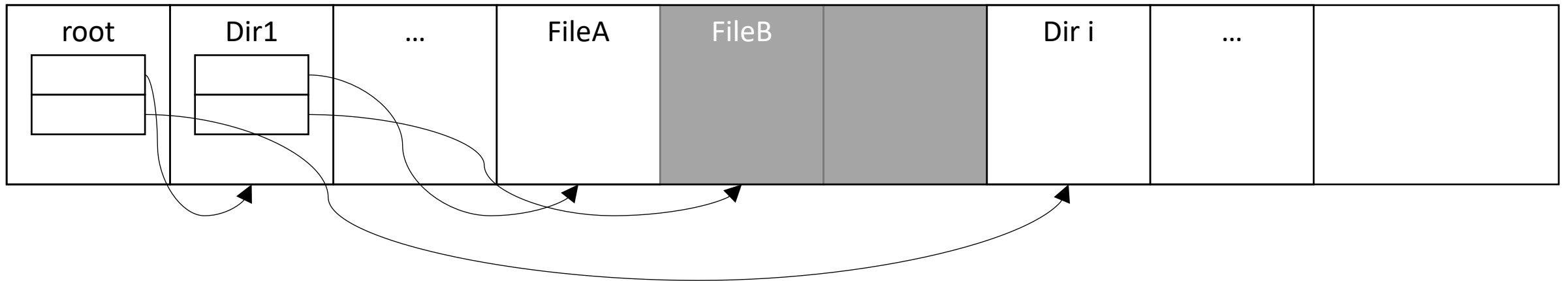
Structure



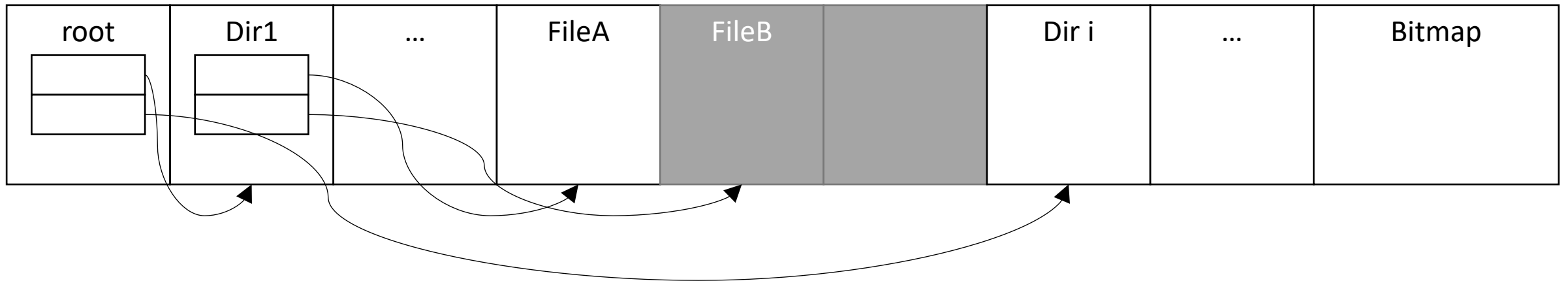
Structure



Structure



Structure



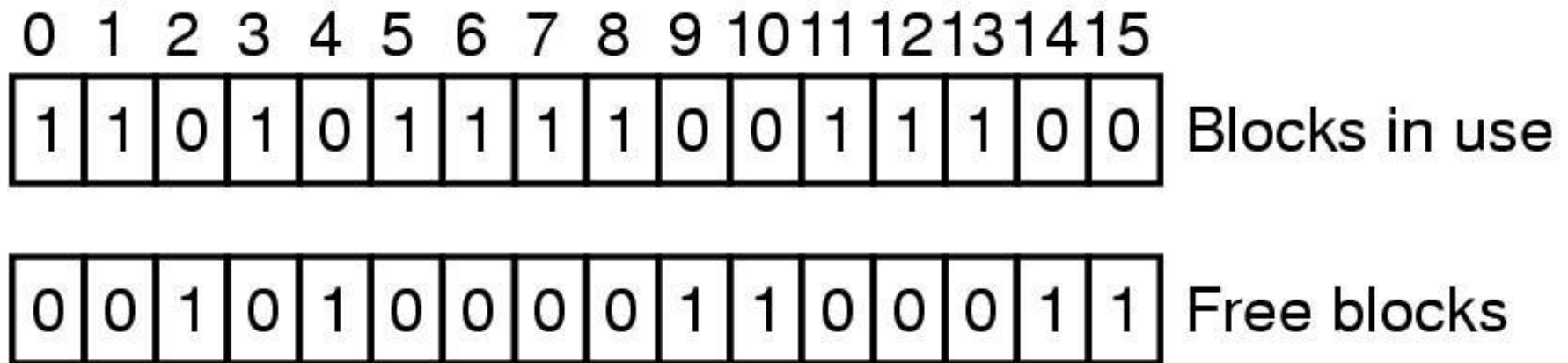
Disk Management

- Manage free (or empty) space using **bitmap**

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---------------|
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | Blocks in use |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | Free blocks |

Disk Management

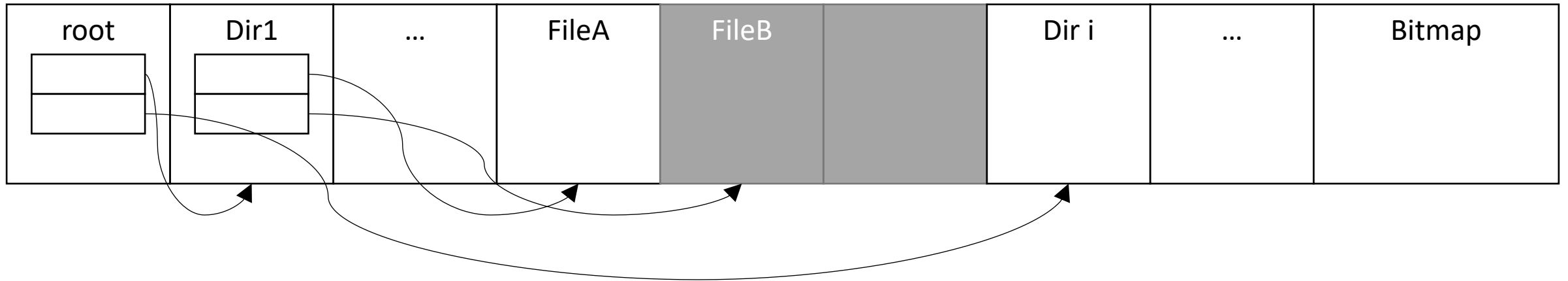
- Manage free (or empty) space using bitmap



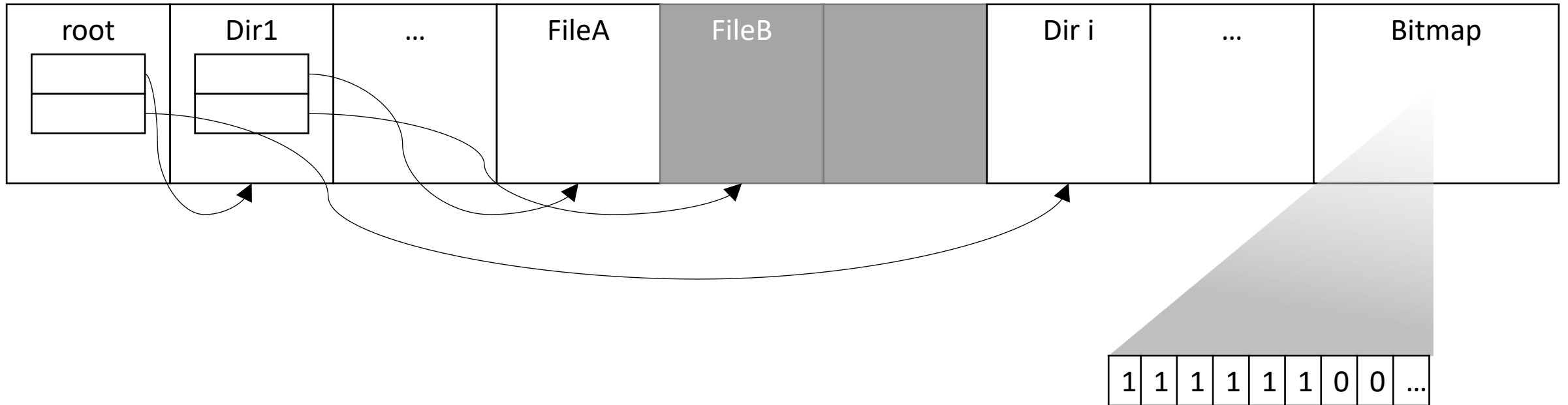
- **Create** a 5MB disk image

```
dd bs=1K count=5K if=/dev/zero of=.disk
```

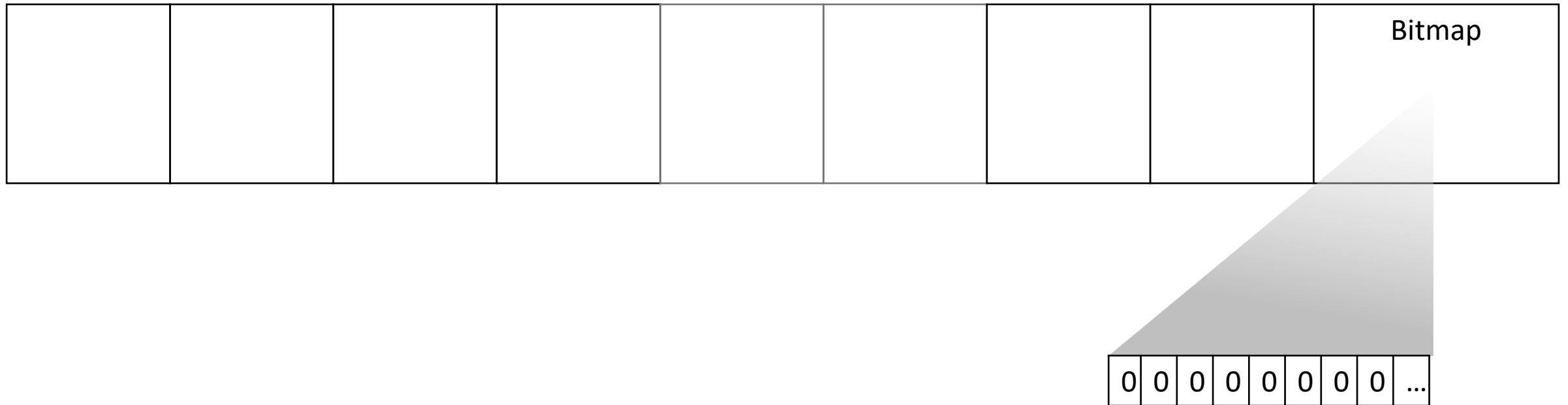
Disk Management

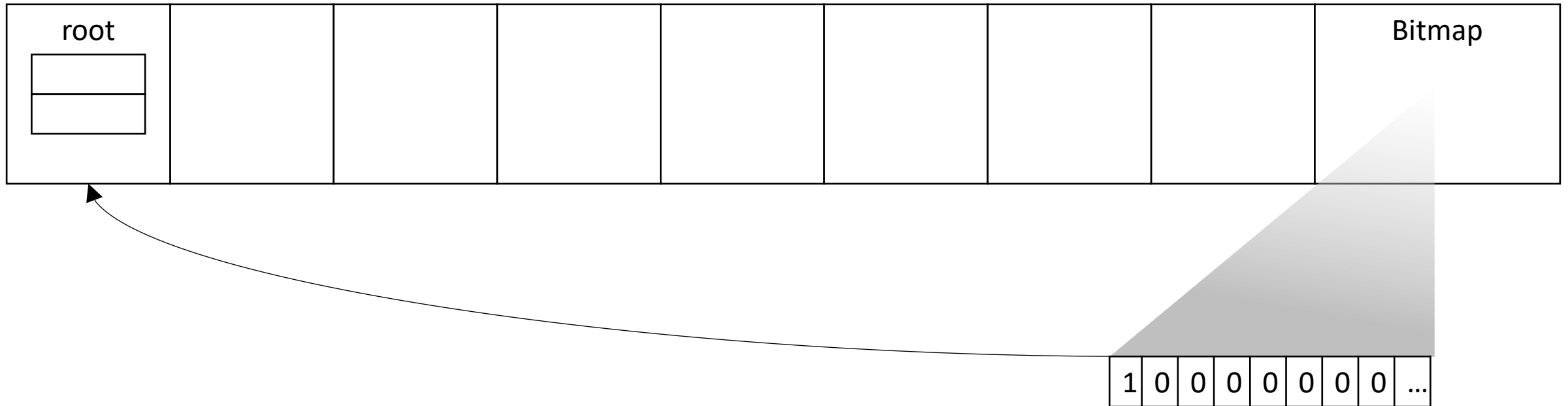


Disk Management

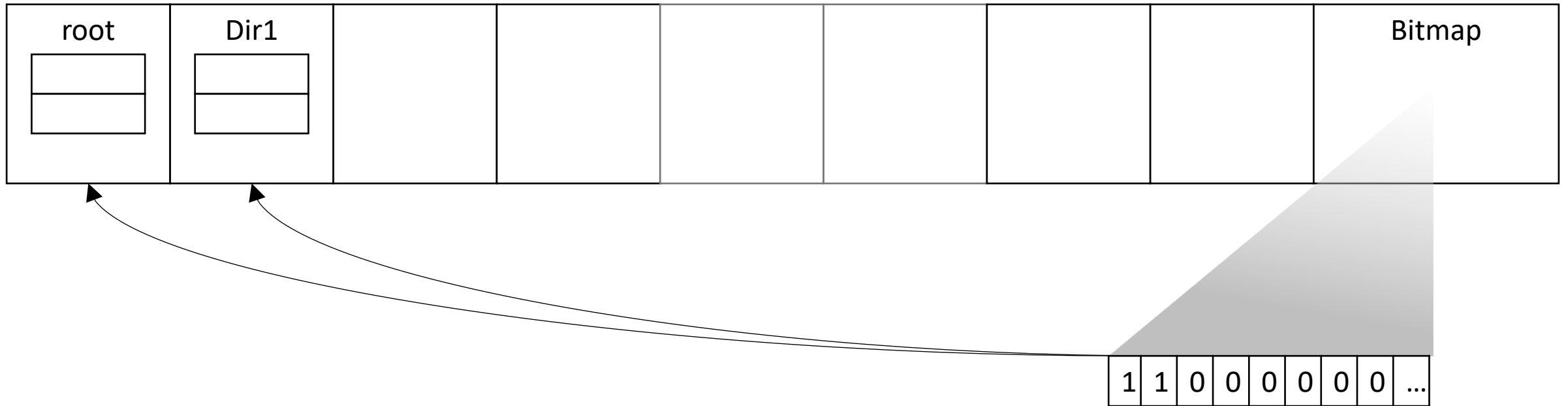


Disk Management

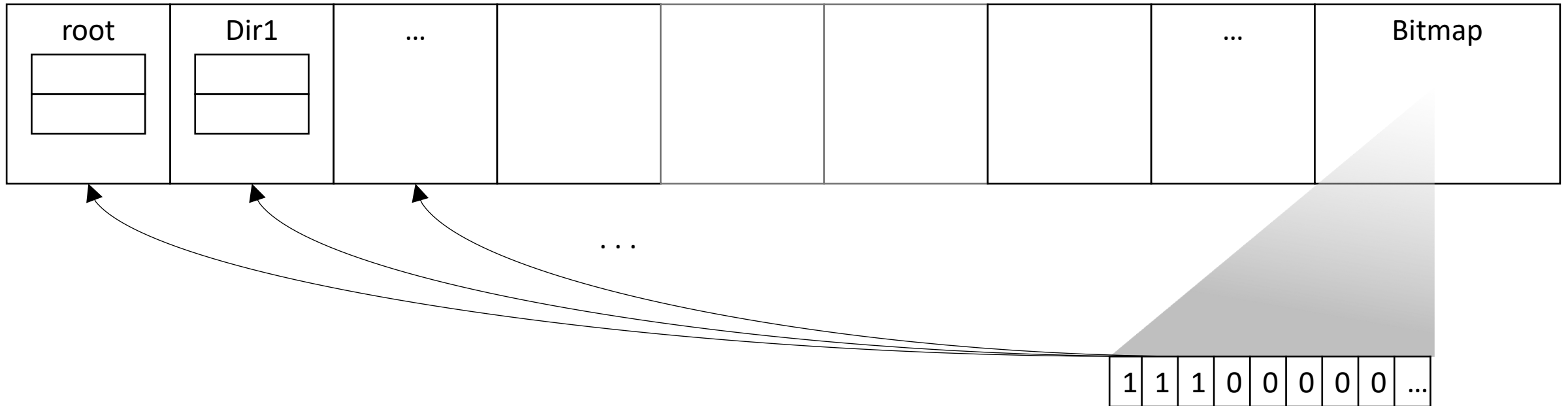




Disk Management



Disk Management



Root Directory

```
struct cs1550_root_directory {  
    int nDirectories;    //How many subdirectories are in the root  
                        //Needs to be less than MAX_DIRS_IN_ROOT  
  
    struct cs1550_directory  
    {  
        char dname[MAX_FILENAME + 1]; //directory name (plus  
space for nul)  
        long nStartBlock; //where the directory block is on disk  
    } directories[MAX_DIRS_IN_ROOT]; //There is an array of these  
};
```

Root Directory

```
struct cs1550_root_directory {  
    int nDirectories;    //How many subdirectories are in the root  
                        //Needs to be less than MAX_DIRS_IN_ROOT  
  
    struct cs1550_directory  
    {  
        char dname[MAX_FILENAME + 1]; //directory name (plus  
space for nul)  
        long nStartBlock; //where the directory block is on disk  
    } directories[MAX_DIRS_IN_ROOT]; //There is an array of these  
};
```

Subdirectories

```
struct cs1550_directory_entry
{
    int nFiles;           //How many files are in this directory.
                          //Needs to be less than MAX_FILES_IN_DIR
    struct cs1550_file_directory
    {
        char fname[MAX_FILENAME + 1];    //filename (plus space for nul)
        char fext[MAX_EXTENSION + 1];    //extension (plus space for nul)
        size_t fsize;                    //file size
        long nStartBlock;                //where the first block is on disk
    } files[MAX_FILES_IN_DIR];          //There is an array of these
};
```

Subdirectories

```
struct cs1550_directory_entry
{
    int nFiles;           //How many files are in this directory.
                        //Needs to be less than MAX_FILES_IN_DIR
    struct cs1550_file_directory
    {
        char fname[MAX_FILENAME + 1];    //filename (plus space for nul)
        char fext[MAX_EXTENSION + 1];    //extension (plus space for nul)
        size_t fsize;                    //file size
        long nStartBlock;                 //where the first block is on disk
    } files[MAX_FILES_IN_DIR];           //There is an array of these
};
```

Files

```
struct cs1550_disk_block {  
    //All the space in the block can be used for actual data  
    //storage.  
    char data[MAX_DATA_IN_BLOCK];  
};
```

Syscalls

- **cs1550_getattr**
- **cs1550_mkdir**
- **cs1550_readdir**
- cs1550_rmdir
- **cs1550_mknod**
- **cs1550_write**
- **cs1550_read**
- cs1550_unlink
- cs1550_truncate
- cs1550_open
- cs1550_flush

Syscalls

- **cs1550_getattr**
- **cs1550_mkdir**
- **cs1550_readdir**
- cs1550_rmdir
- **cs1550_mknod**
- **cs1550_write**
- **cs1550_read**
- cs1550_unlink
- cs1550_truncate
- cs1550_open
- cs1550_flush

No delete calls need to be written so
you don't need to solve fragmentation

When there is no space left, return an error

Requirements and submission

- Well-commented cs1550.c
- Rubric

| Item | Grade |
|-----------------------------|-------|
| cs1550_getattr | 15% |
| cs1550_mkdir | 15% |
| cs1550_readdir | 15% |
| cs1550_mknod | 15% |
| cs1550_write | 15% |
| cs1550_read | 15% |
| File System works correctly | 10% |