CS 1550

Week 12 – Project 4

Teaching Assistant

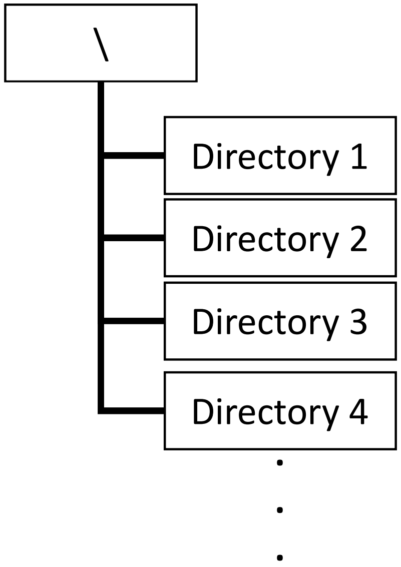
Xiaoyu(Veronica) Liang

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**

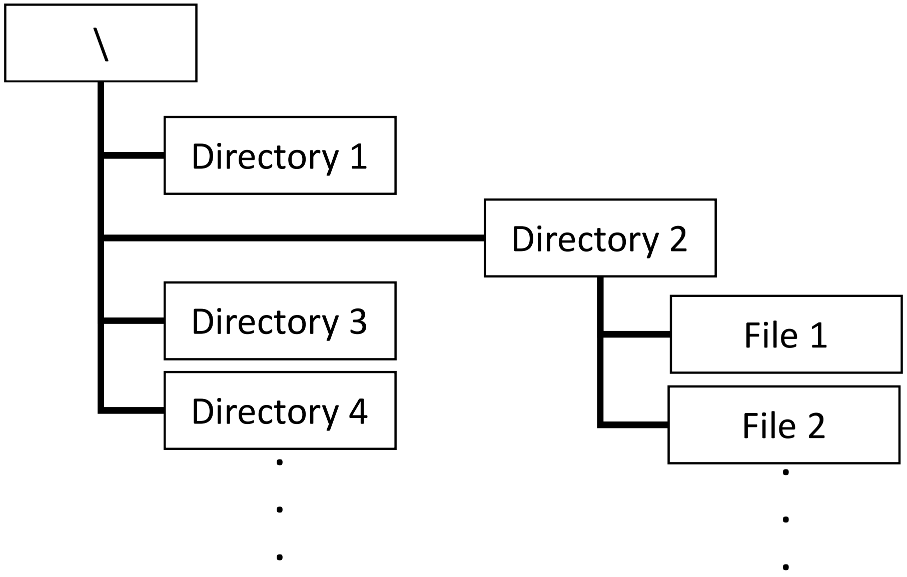
File System

- 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.
  - 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.

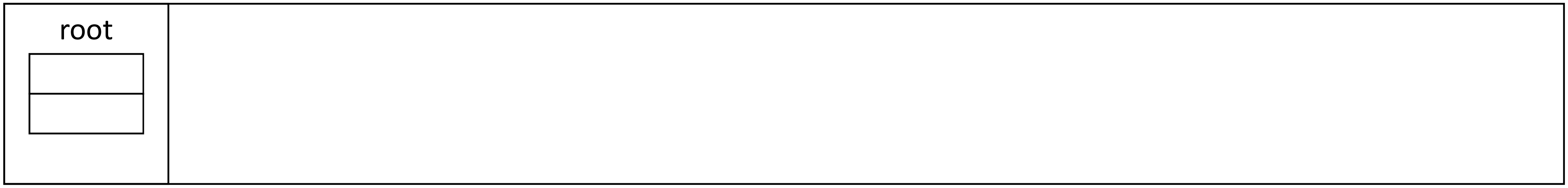
File System

- 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**.

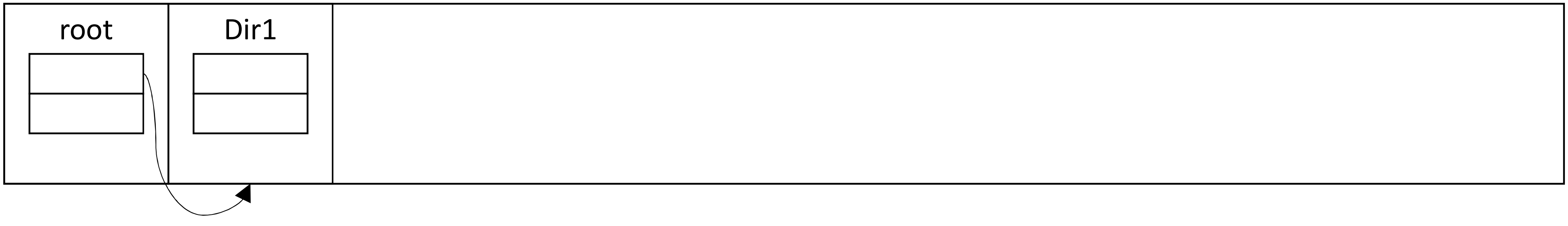
File System

- 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**.

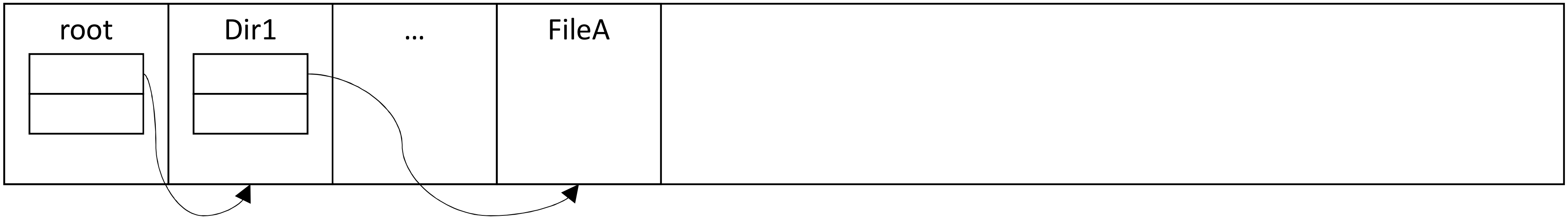
Structure



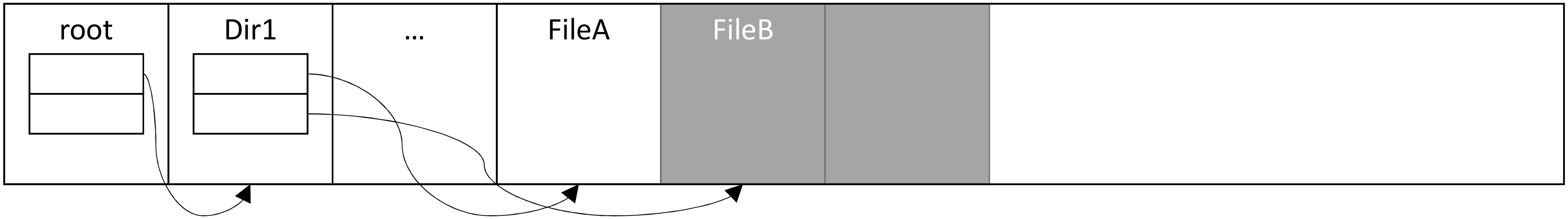
Structure



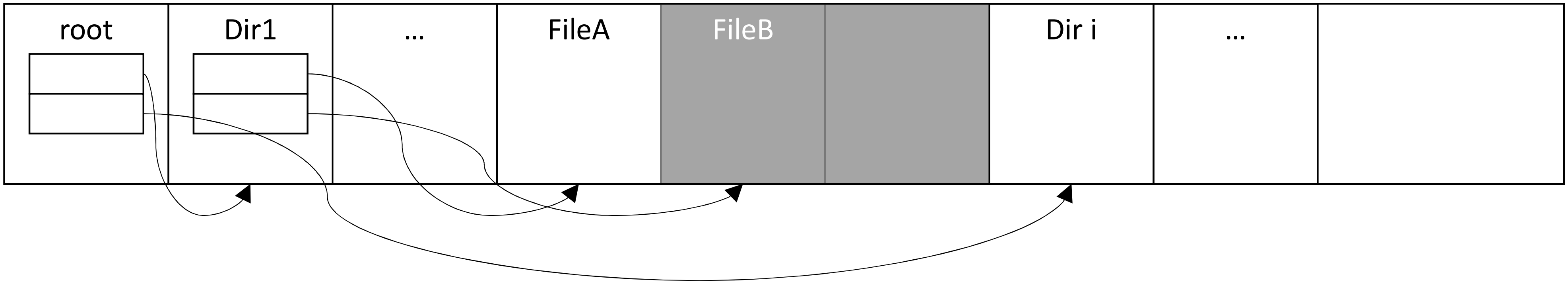
Structure



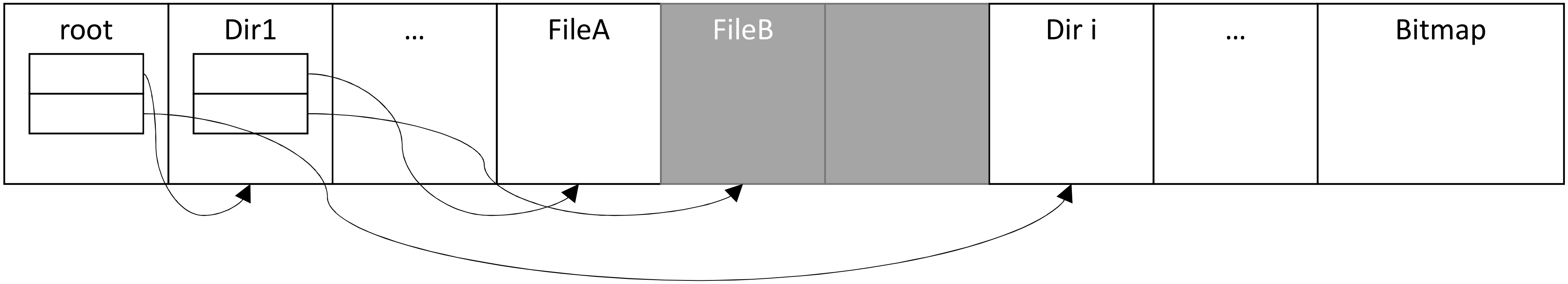
Structure



Structure

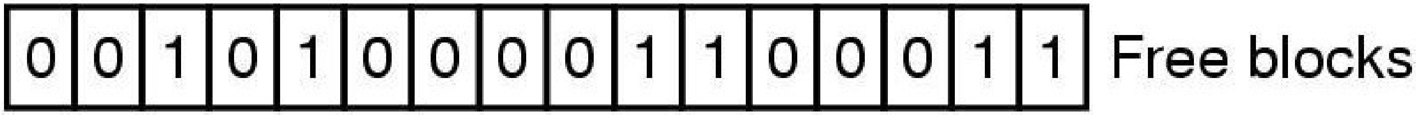


Structure



Disk Management

- Manage free (or empty) space using bitmap

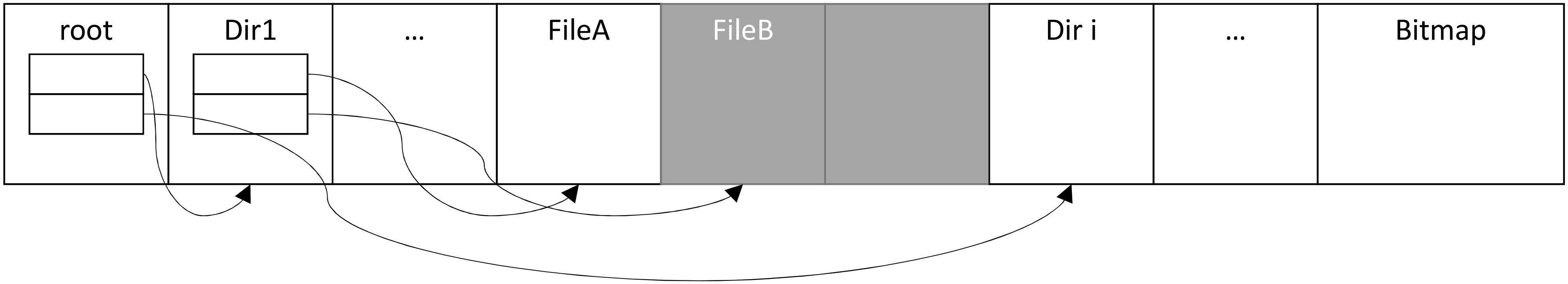


(a)

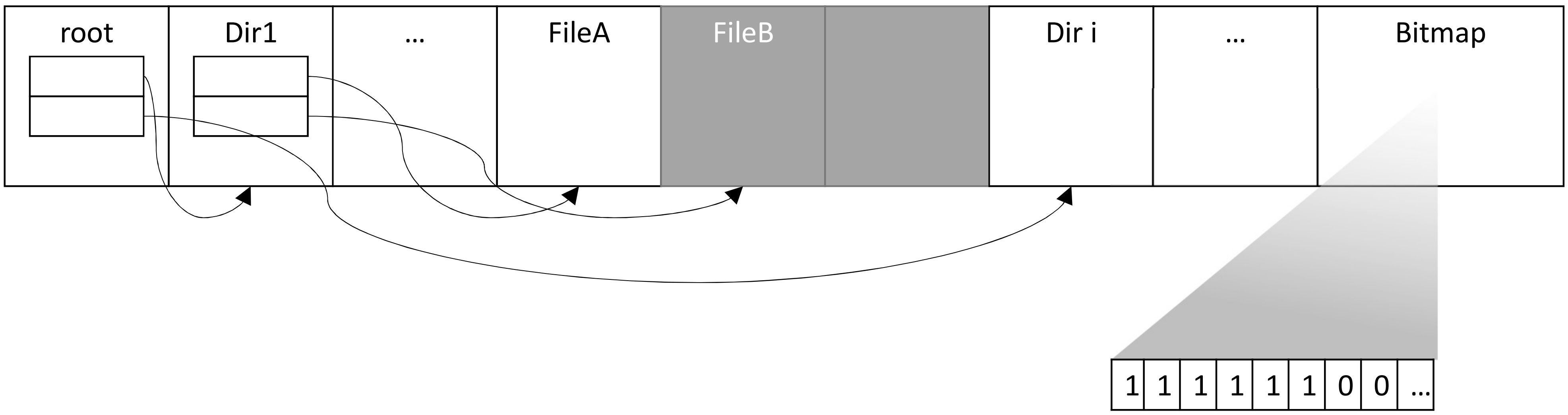
- **Create** a 5MB disk image

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

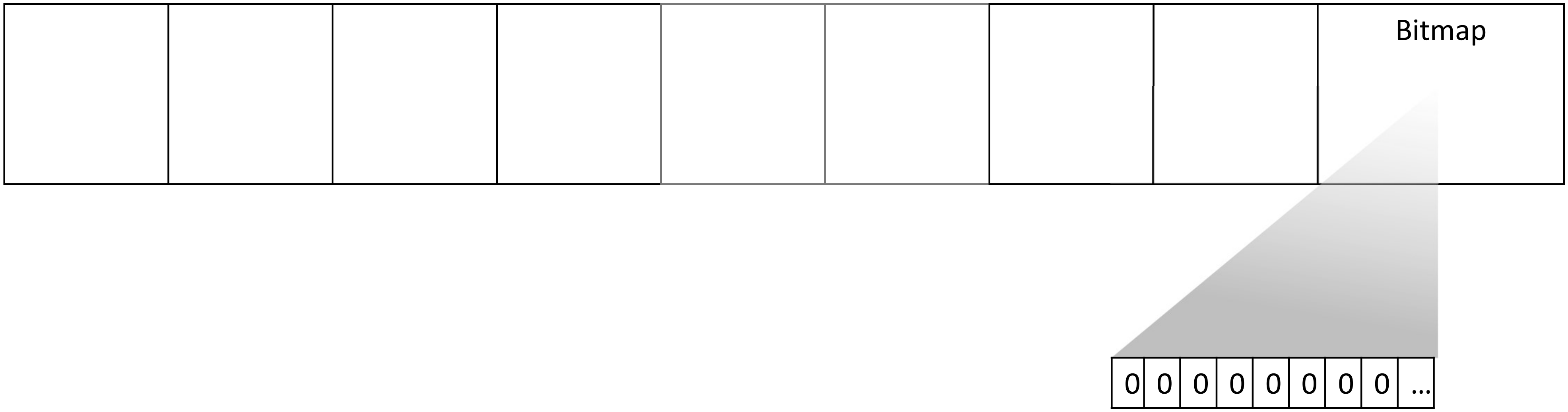
Disk Management



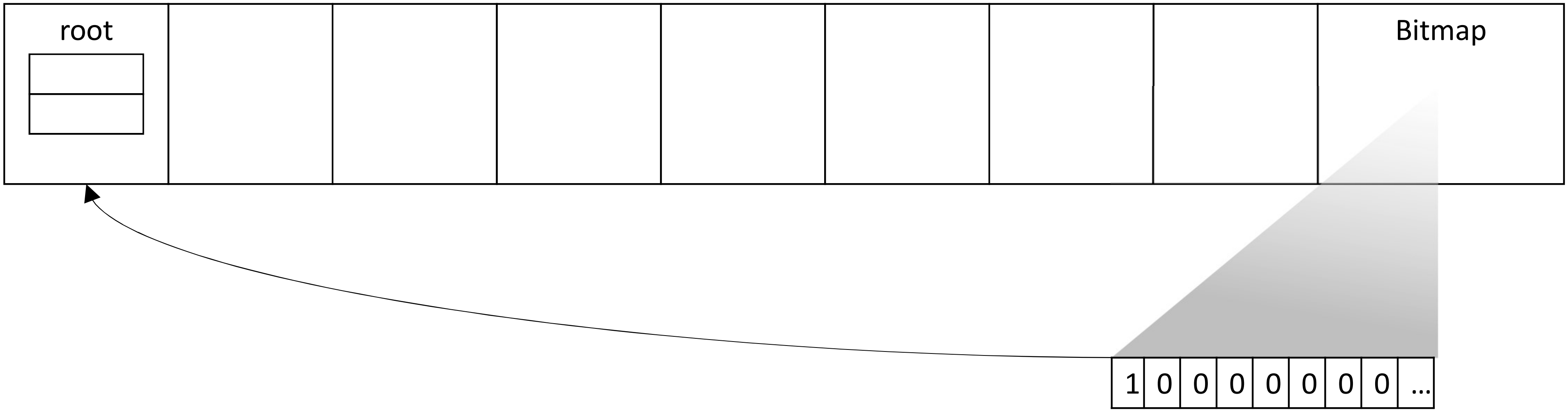
Disk Management



Disk Management



Disk Management

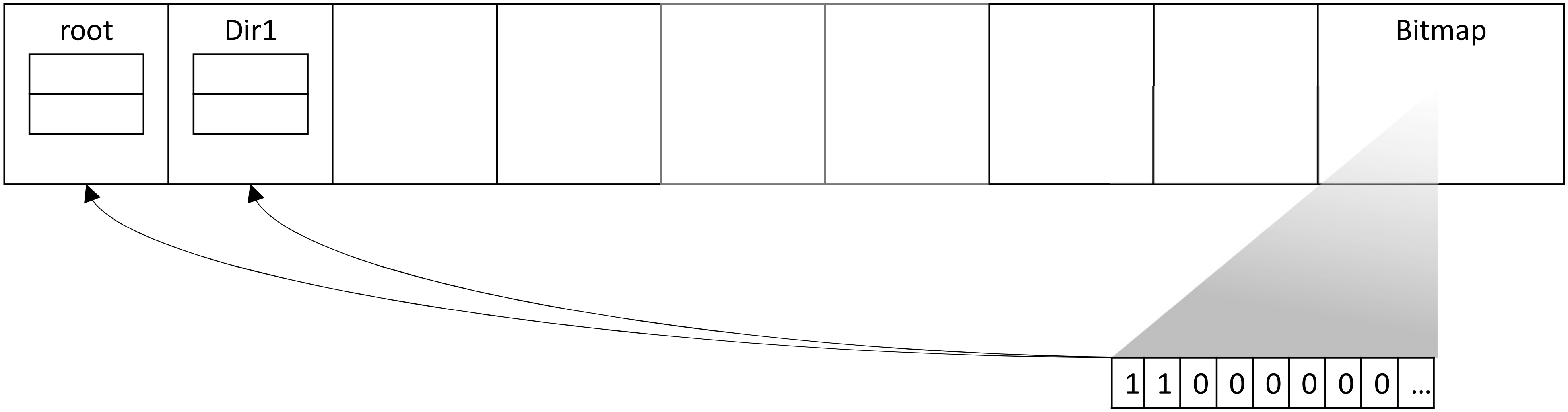


root

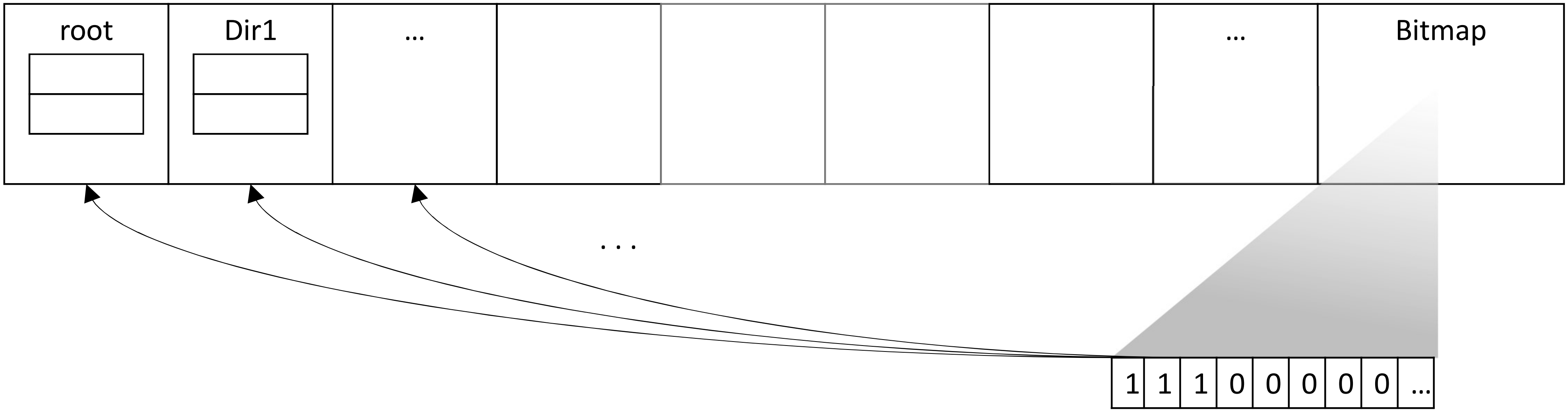
Bitmap

1	0	0	0	0	0	0	0	…

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
};
```

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%