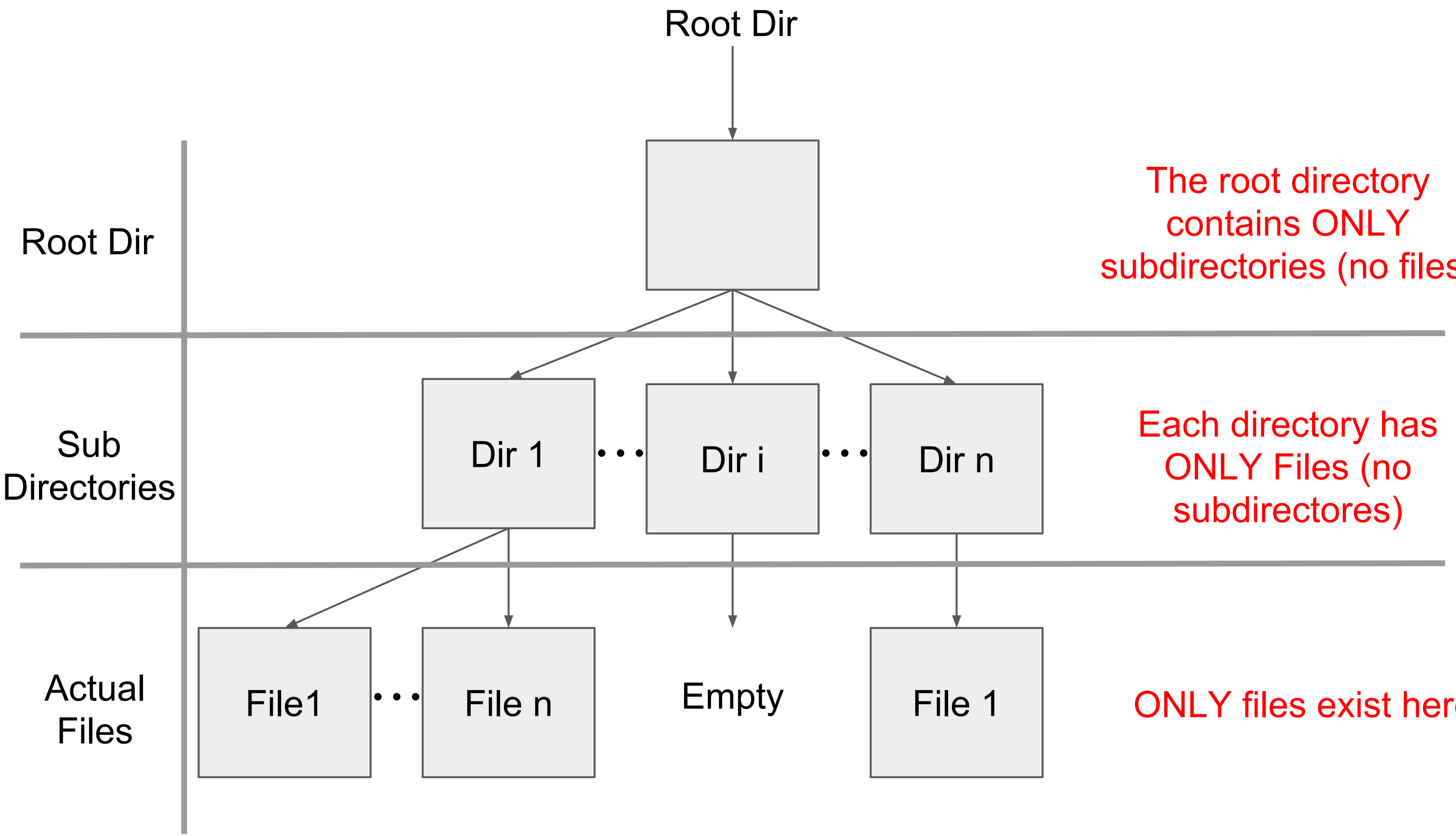
Discussion 4/12/19

Assignment 5 - File System

- Implement a very minimal file system.
- Two layer directory hierarchy, no more than that.
- You will be using FUSE to create it.
- Deadlines:
  - Partners due - YESTERDAY
  - Directories portion - April 23rd
  - Entire project - May 1st

FS Structure



Root Dir

Root Dir

Sub Directories

Actual Files

Dir 1 • • • Dir i • • • Dir n

File1 • • • File n

Empty

File 1

The root directory contains ONLY subdirectories (no files)

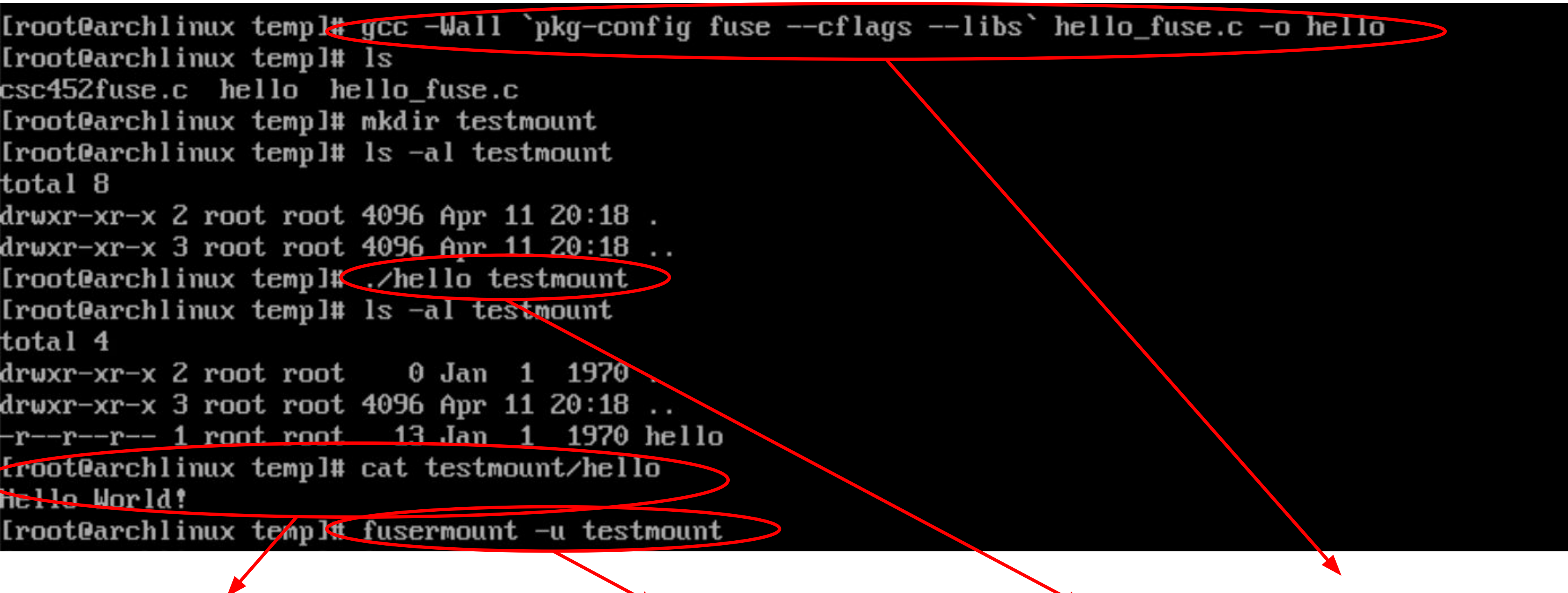
Each directory has ONLY Files (no subdirectores)

ONLY files exist here

FUSE

- A Linux kernel extension for creating file systems that exist in user space.
- Allows us to implement functions that handle system calls on our “fake” user space file system.
- To install on VM, run:
    - **pacman -Sy**
    - **pacman -S fuse tmux**
- Then copy files onto VM:
    - **scp USERNAME@lectura.cs.arizona.edu:~jmisurda/original/csc452fuse.c .**
    - **scp USERNAME@lectura.cs.arizona.edu:~jmisurda/original/hello_fuse.c .**
- hello

FUSE



```
[root@archlinux temp]# gcc -Wall `pkg-config fuse --cflags --libs` hello_fuse.c -o hello
[root@archlinux temp]# ls
csc452fuse.c  hello  hello_fuse.c
[root@archlinux temp]# mkdir testmount
[root@archlinux temp]# ls -al testmount
total 8
drwxr-xr-x 2 root root 4096 Apr 11 20:18 .
drwxr-xr-x 3 root root 4096 Apr 11 20:18 ..
[root@archlinux temp]# ./hello testmount
[root@archlinux temp]# ls -al testmount
total 4
drwxr-xr-x 2 root root    0 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Apr 11 20:18 ..
-r--r--r-- 1 root root   13 Jan  1  1970 hello
[root@archlinux temp]# cat testmount/hello
Hello World!
[root@archlinux temp]# fusermount -u testmount
```

This cat is handled by the function in the hello FS

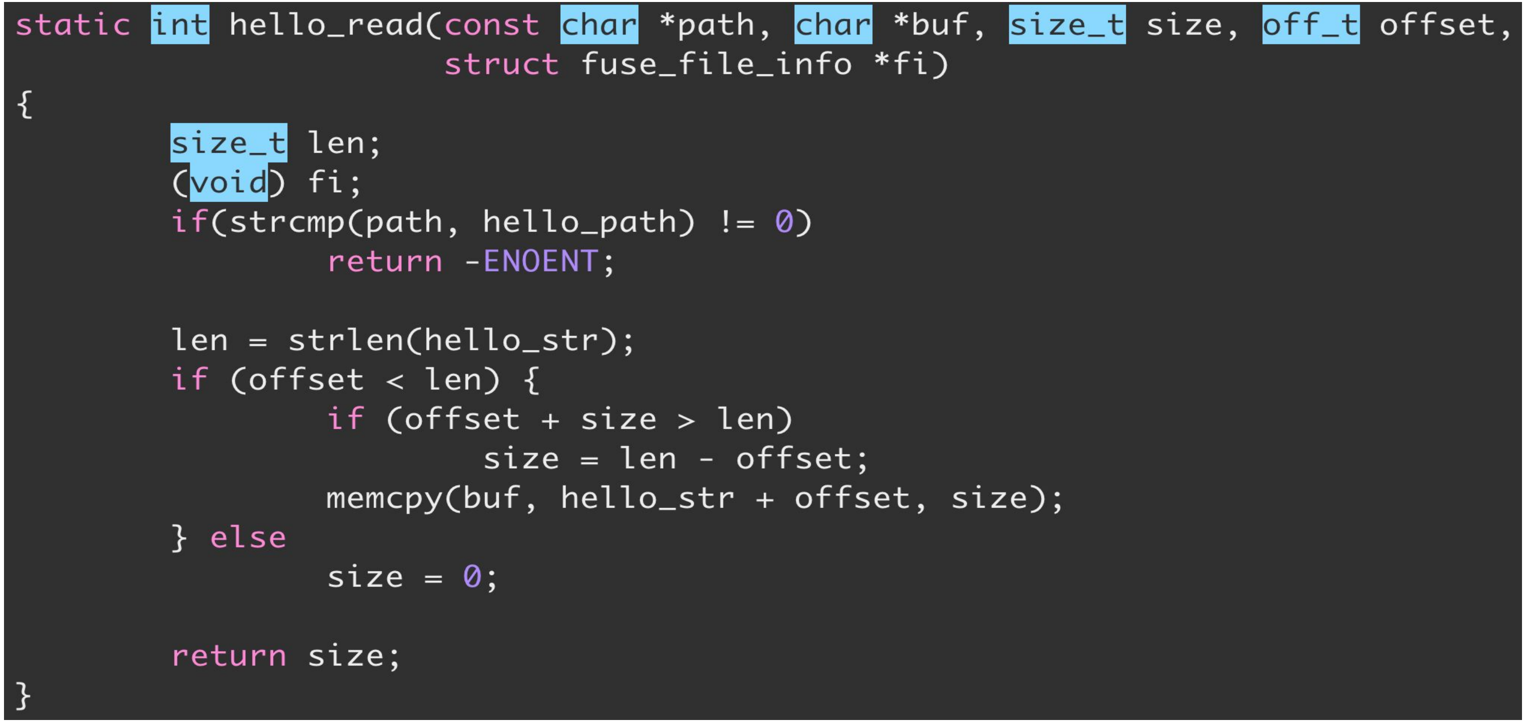
Unmounts file system

Causes FUSE to mount FS in the testmount directory (mount point)

Compilation command

FUSE - Implementation

- Let us look at how that last example worked:

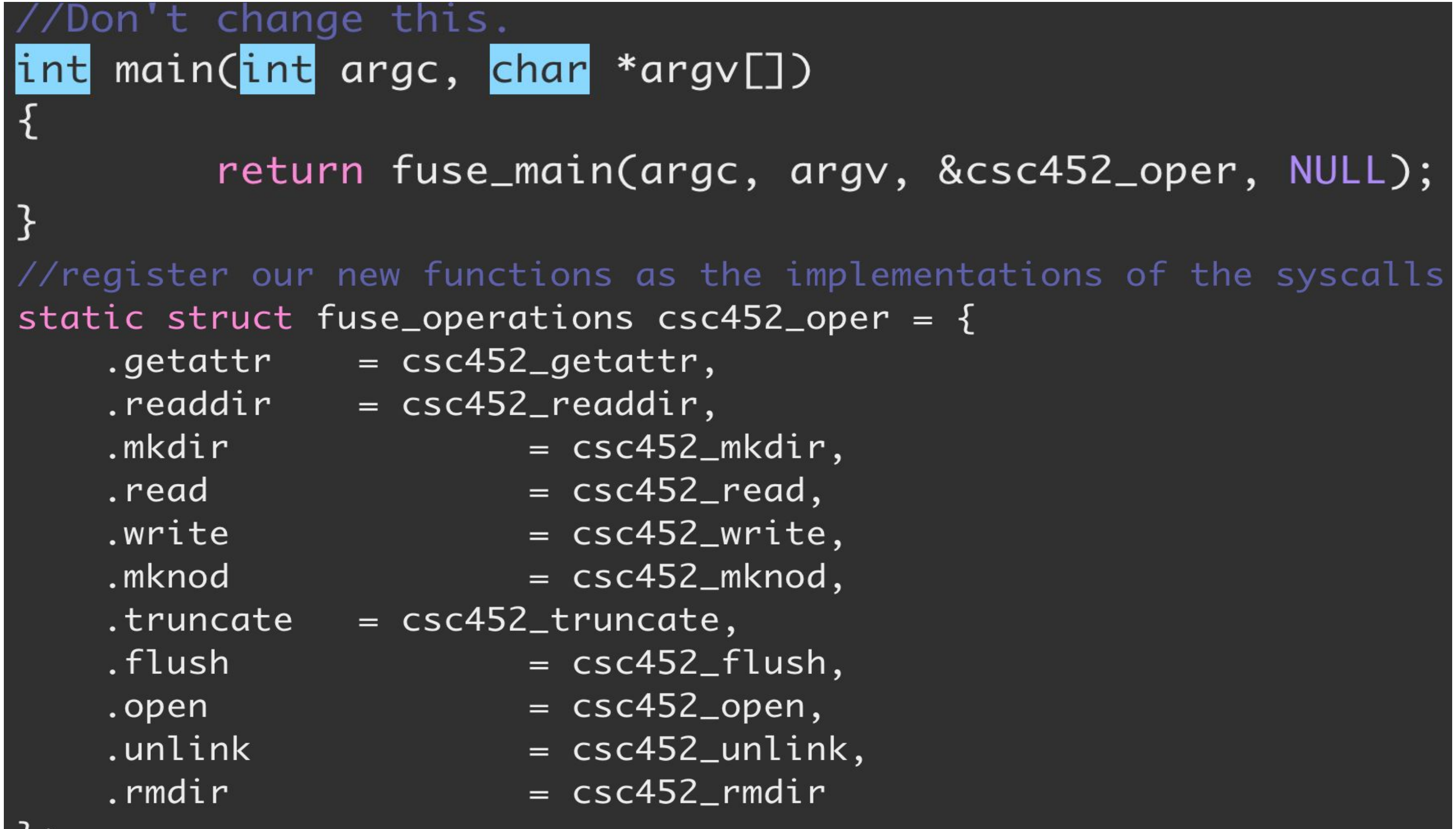


```
static int hello_read(const char *path, char *buf, size_t size, off_t offset,
                      struct fuse_file_info *fi)
{
        size_t len;
        (void) fi;
        if(strcmp(path, hello_path) != 0)
                return -ENOENT;

        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;
}
```

FUSE - Implementation



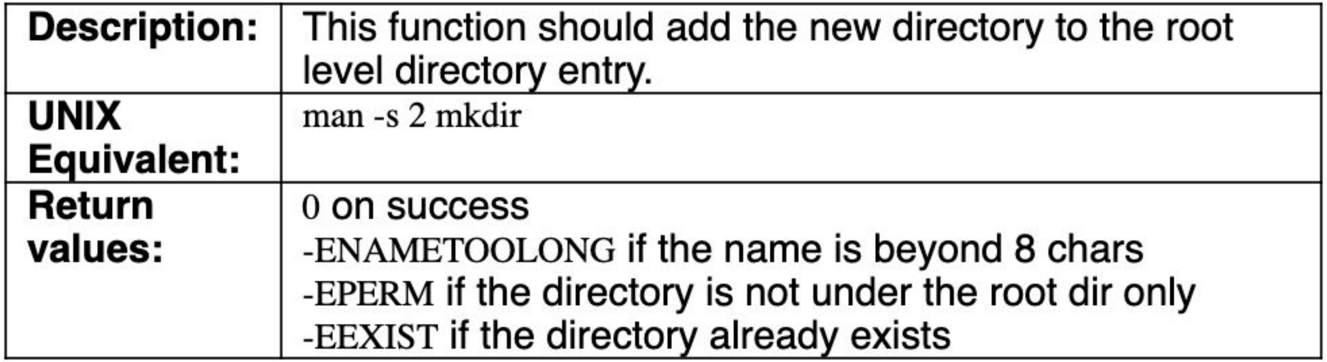
```
//Don't change this.
int main(int argc, char *argv[])
{
        return fuse_main(argc, argv, &csc452_oper, NULL);
}
//register our new functions as the implementations of the syscalls
static struct fuse_operations csc452_oper = {
    .getattr    = csc452_getattr,
    .readdir    = csc452_readdir,
    .mkdir              = csc452_mkdir,
    .read               = csc452_read,
    .write              = csc452_write,
    .mknod              = csc452_mknod,
    .truncate   = csc452_truncate,
    .flush              = csc452_flush,
    .open               = csc452_open,
    .unlink             = csc452_unlink,
    .rmdir              = csc452_rmdir
};
```

Functions to Implement

- On this assignment, your job is to fill out csc452fuse.c
- File system is implemented using a single file, managed by the real file system in the same directory as your FS. This file should keep track of the directories and the file data.
- We will treat the disk as if it has 512-byte blocks.
- Don't change / ignore these functions in the starter code:
  - csc452_open, csc452_flush, csc452_truncate
- You will be implementing the following functions to handle system calls:
  - csc452_unlink, csc452_read, csc452_write, csc452_mknod, csc452_rmdir, csc452_readdir, csc452_getattr, csc452_mkdir

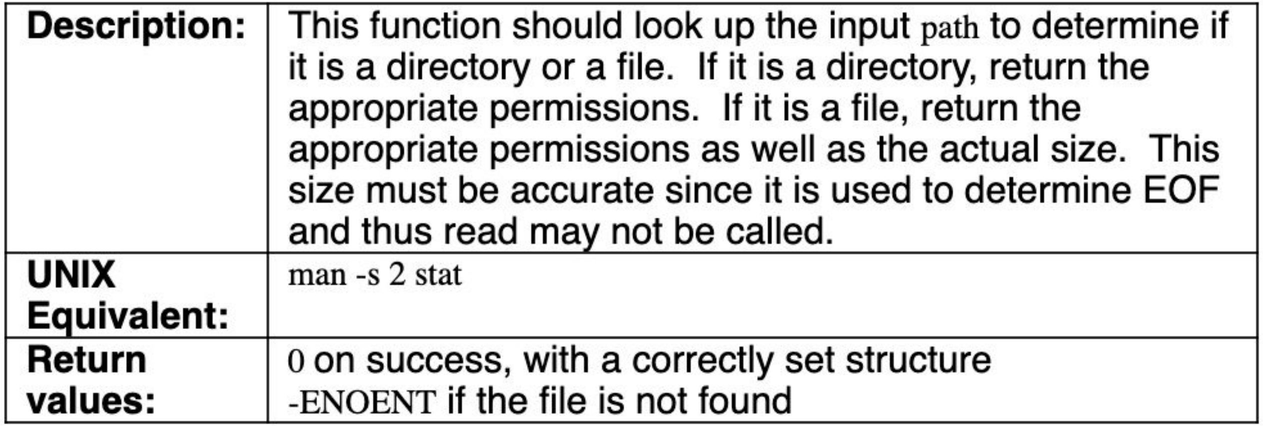
Functions to Implement (Cont.)

csc452_mkdir



Description:	This function should add the new directory to the root level directory entry.
UNIX Equivalent:	man -s 2 mkdir
Return values:	0 on success -ENAMETOOLONG if the name is beyond 8 chars -EPERM if the directory is not under the root dir only -EEXIST if the directory already exists

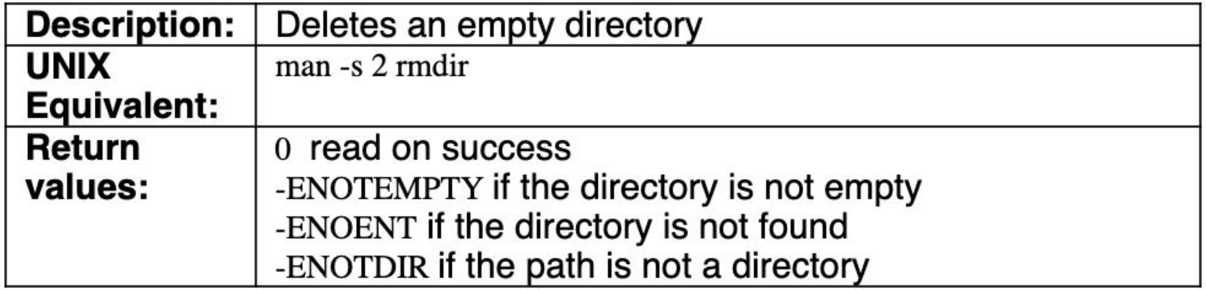
csc452_getattr



Description:	This function should look up the input path to determine if it is a directory or a file. If it is a directory, return the appropriate permissions. If it is a file, return the appropriate permissions as well as the actual size. This size must be accurate since it is used to determine EOF and thus read may not be called.
UNIX Equivalent:	man -s 2 stat
Return values:	0 on success, with a correctly set structure -ENOENT if the file is not found

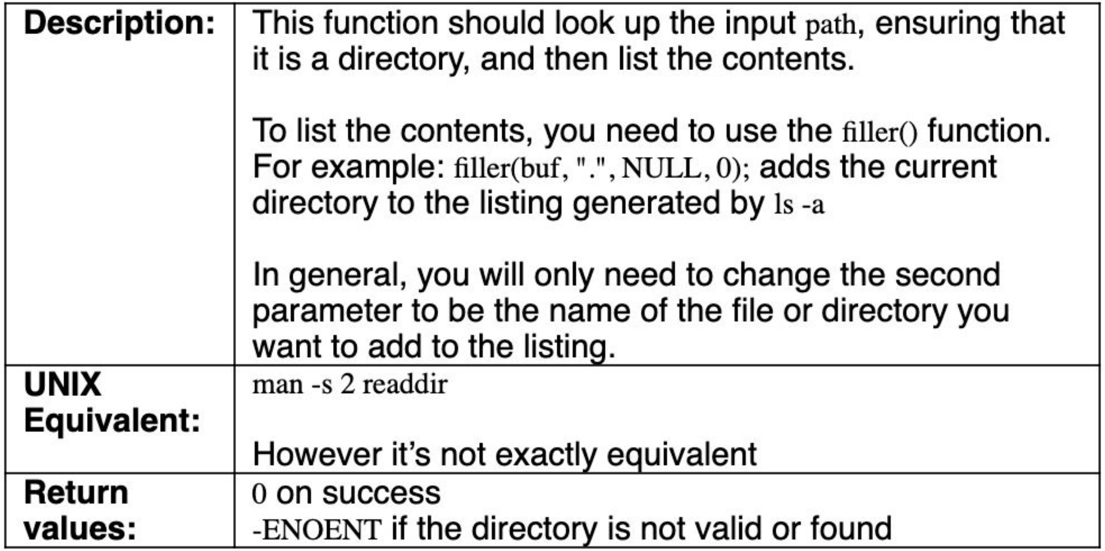
Functions to Implement (Cont.)

csc452_rmdir



Description:	Deletes an empty directory
UNIX Equivalent:	man -s 2 rmdir
Return values:	0 read on success -ENOTEMPTY if the directory is not empty -ENOENT if the directory is not found -ENOTDIR if the path is not a directory

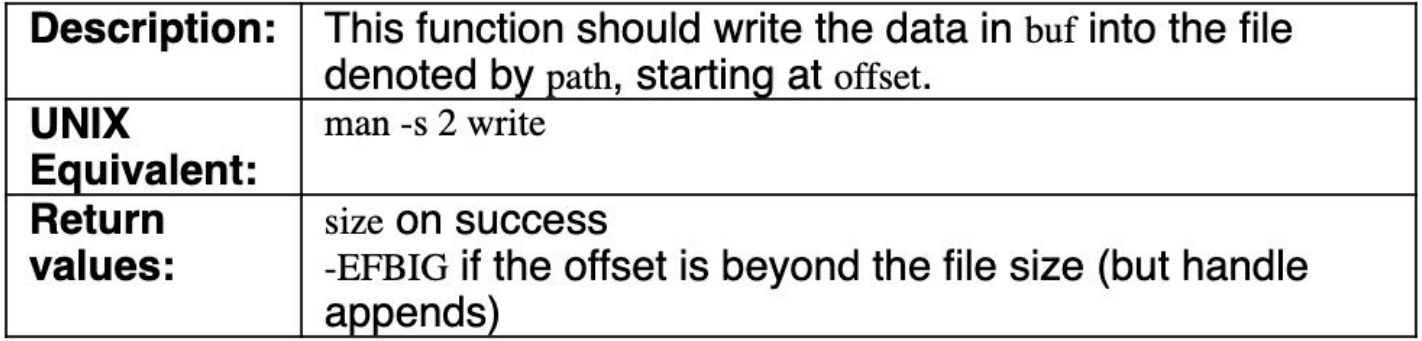
csc452_readdir



Description:	This function should look up the input path, ensuring that it is a directory, and then list the contents. To list the contents, you need to use the filler() function. For example: filler(buf, ".", NULL, 0); adds the current directory to the listing generated by ls -a In general, you will only need to change the second parameter to be the name of the file or directory you want to add to the listing.
UNIX Equivalent:	man -s 2 readdir However it's not exactly equivalent
Return values:	0 on success -ENOENT if the directory is not valid or found

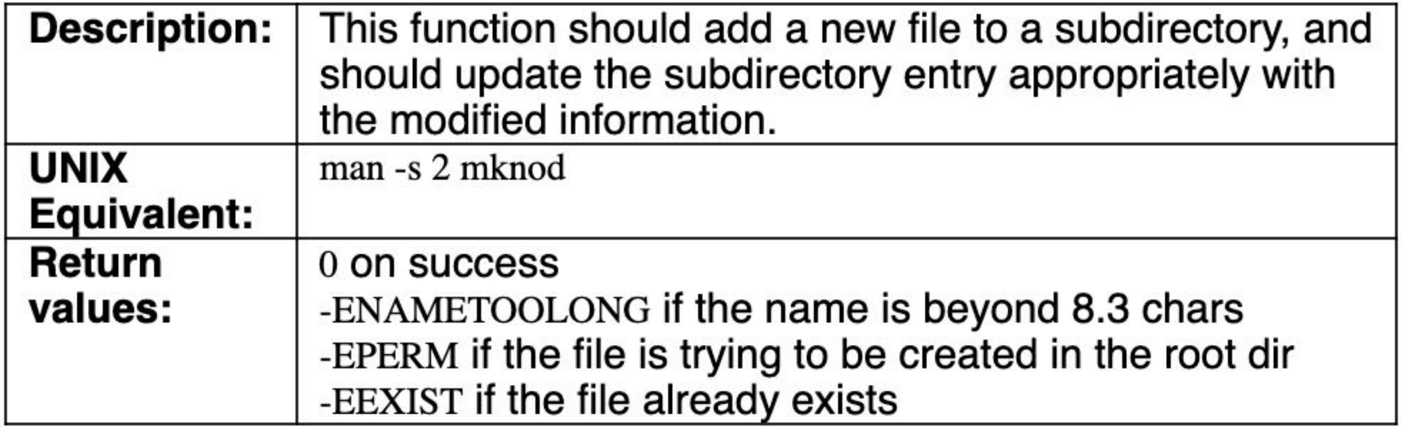
Functions to Implement (Cont.)

csc452_write



Description:	This function should write the data in buf into the file denoted by path, starting at offset.
UNIX Equivalent:	man -s 2 write
Return values:	size on success -EFBIG if the offset is beyond the file size (but handle appends)

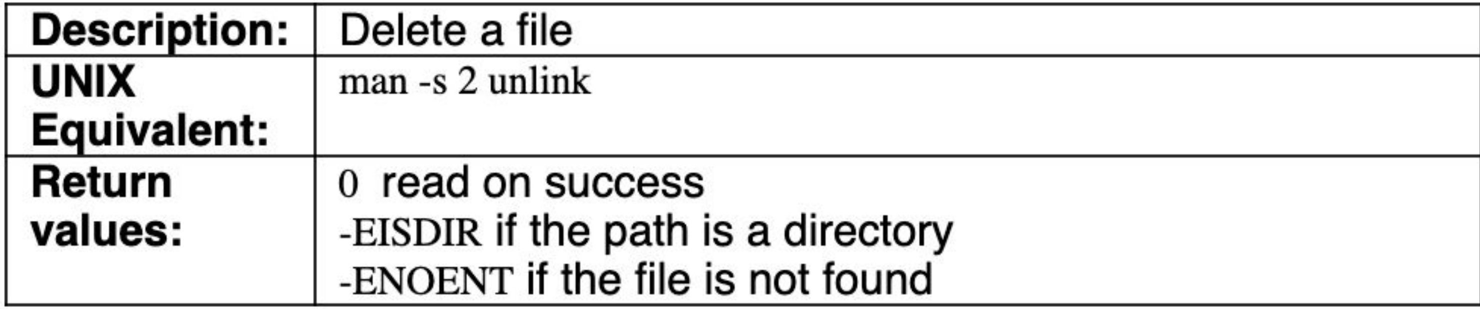
csc452_mknod



Description:	This function should add a new file to a subdirectory, and should update the subdirectory entry appropriately with the modified information.
UNIX Equivalent:	man -s 2 mknod
Return values:	0 on success -ENAMETOOLONG if the name is beyond 8.3 chars -EPERM if the file is trying to be created in the root dir -EEXIST if the file already exists

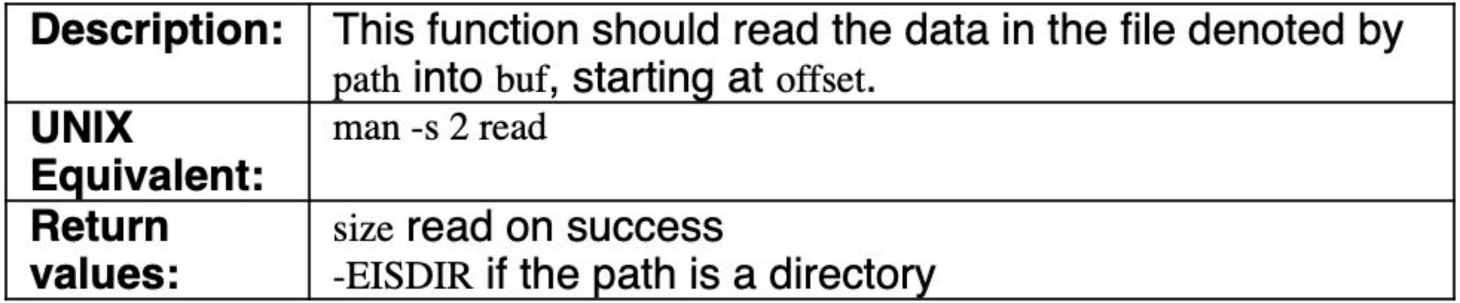
Functions to Implement (Cont.)

csc452_unlink



Description:	Delete a file
UNIX Equivalent:	man -s 2 unlink
Return values:	0 read on success -EISDIR if the path is a directory -ENOENT if the file is not found

csc452_read



Description:	This function should read the data in the file denoted by path into buf, starting at offset.
UNIX Equivalent:	man -s 2 read
Return values:	size read on success -EISDIR if the path is a directory

Specifics (to be continued…)

- Next week we will get into the specifics of how to implement these things.

Quiz Time!!!

- 25 mins
- 5 questions