

CS 1550

Week 13

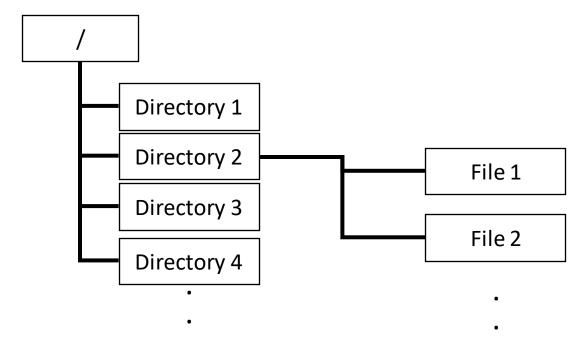
_

Project 4 cont.

TA: Henrique Potter

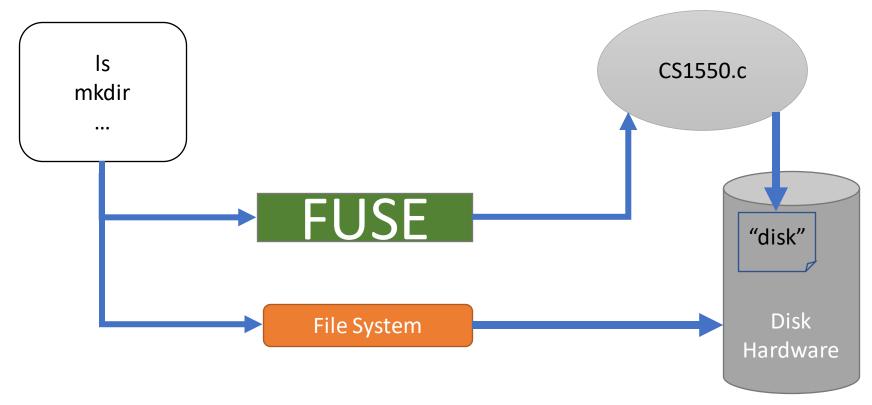
Project 4: File System

- 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.



Create a file as "disk"

- We need to create a file as the "disk" for our file system. All metadata and file data in our file system will be stored in this "disk".
 - Create a 5MB file: dd bs=1K count=5K if=/dev/zero of=.disk



Syscalls

This project requires:

- cs1550_getattr
 - 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
- cs1550_mkdir
 - This function should add the new directory to the root level
- cs1550_mknod
 - This function should add a new file to a subdirectory

Initially, the ".disk" created contains all zeros

Block 0	Block 1	Block 2	Block 3	Block 4	•••

The block 0 is for root.

Block 0	Block 1	Block 2	Block 3	Block 4	•••				
root	_								
struct c	s1550_root_dir	rectory							
10	<pre>{ long lastAllocatedBlock; //The number of the last allocated block = 0</pre>								
in	t nDirectories	•			= ()				
_	//Needs to be less than MAX_DIRS_IN_ROOT struct cs1550_directory dname nStartBlock								
{	<pre>{ char dname[MAX_FILENAME + 1]; //directory name (plus space for nul) \[\0\0 \] 0</pre>								
}	long nSta directories[M/	-			ctory block is on d ray of these	isk	\0\0	0	
} ;	_	_ _	_						

Now let's create a directory DirA: We need to parse and validate the path "/DirA"

Block 0	Block 1	Block 2	Block 3	Block 4	•••
root					

/DirA:

"/": root directory of our file system

"DirA": the subdirectory to be created inside root directory

Now let's create a directory DirA: We need to parse and validate the path "/DirA"

Block (D Block 1	Block 2	Block 3	Block 4				
root								
{	cs1550_root_d	•	//The numbe	er of the la	st allocated b	lock = 0		_
	nt nDirectori truct cs1550_		nany subdire s to be less			= 0	dname nSta	artBlock
{	long nS	tartBlock;	//whe	re the dire	ctory block is	s space for nul on disk) \0\0 0 \0\0 0	
};	all ecrol tes[MAX_DIRS_IN_F	(OOI], //INE	are 12 an al.	ray or these	DirA does not e	oxist	

DirA does not exist. We can proceed.

Now let's create a directory DirA: We need to allocate a block for directory metadata

Block 0	Block 1	Block 2	Block 3	Block 4	•••		
root							
{ long int	nDirectories	edBlock; ; //How m //Needs	any subdire	ctories are	st allocated block = 0 in the root = 0 IRS_IN_ROOT		
{	long nSta	e[MAX_FILEN rtBlock;	//whe	re the dire	tory name (plus space for nul) ctory block is on disk ray of these	\0\0 0 \0\0 0	rtBlock

Now let's create a directory DirA: We need to allocate a block for directory metadata

Block 0	Block 1	Block 2	Block 3	Block 4			
root							
{ long		edBlock;		ctories are	st allocated blo in the root IRS IN ROOT	5, 55 115 1	l use block 1, and is number by 1
stru	ict cs1550_di			_			dname nStartBlock
{	char dnam	ne[MAX_FILEN	AME + 1];	//direc	tory name (plus	space for nul	\0\0 0
ો ત	long nSta rectories[MA	-			ctory block is o	on disk	\0\0 0
};	i ectorites[MA	ı∨_∩τ <i>µ</i> ,2 [_] ΤI <i>I</i>	, // me	i E 12 ali al.	i ay or these		

Now let's create a directory DirA: <u>Update the root metadata</u>

} ;

Block 0	Block 1	Block 2	Block 3	Block 4	•••
root	DirA				

Now let's create a directory DirA: <u>Update the directory metadata</u>

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA				

```
struct cs1550_directory entry
                                     //How many files are in this directory.
     int nFiles;
                                                                               = 0
                                     //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 nIndexBlock;
                                           //where the index block is on disk
     } files[MAX FILES IN DIR];
                                           //There is an array of these
};
```

Fname nIndexBlock

\0\0	0
\0\0	0

Now let's create a directory DirA: Save changes to "disk"

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA				

The creation of "DirA" modifies data inside block 0 and block 1, so we need to save these changes into "disk".

Now let's create a file "FileA.txt" inside DirA: Parse and validate the path "/DirA/FileA.txt"

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA				

/DirA/FileA.txt:

"/" : root directory of our file system

"DirA" : the subdirectory that contains our target file

"FileA" : the target file name

"txt": the target file extension

Now let's create a file "FileA.txt" inside DirA: Parse and validate the path "/DirA/FileA.txt"

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA				

/DirA exists

Now let's create a file "FileA.txt" inside DirA: Parse and validate the path "/DirA/FileA.txt"

struct cs1550 directory entry

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA				

```
int nFiles;
                                     //How many files are in this directory.
                                                                                = 0
                                     //Needs to be less than MAX_FILES_IN_DIR
      struct cs1550 file directory
                                                                                                     nIndexBlock
                                                                                            Fname
            char fname[MAX FILENAME + 1];
                                          //filename (plus space for nul)
             char fext[MAX EXTENSION + 1];
                                               //extension (plus space for nul)
                                                                                             \0\0..
             size_t fsize;
                                            //file size
            long nIndexBlock;
                                            //where the index block is on disk
                                                                                              \0\0...
      } files[MAX FILES IN DIR];
                                            //There is an array of these
};
```

/DirA has no file named FileA.txt, So we can proceed.

Now let's create a file "FileA.txt" inside DirA: Allocate two blocks for FileA, and update root & dirA & FileA metadata

};

В	lock 0	Block 1	Block 2	Block 3	Block 4			
	root	DirA	FileA	FileA's initial data block				
str {	long		edBlock;			ast allocated block = 1 → 3	3, use block 2	2 & 3 for FileA
	int n	Directories		nany subdired s to be less		e in the root DIRS_IN_ROOT = 1		
	struc	t cs1550_di	rectory				dname	nStartBlock
	{					tory name (plus space for	nul) DirA	1
	} dir		= = = = = = = = = = = = = = = = = = =			ectory block is on disk eray of these	\0\0	0

Now let's create a file "FileA.txt" inside DirA: Allocate two blocks for FileA, and update root & dirA & FileA metadata

};

	Block 0	Block 1	Block 2	Block 3	Block 4				
	root	DirA	FileA	FileA's initial data block					
,	{	550_directory	_entry			in this directory. an MAX_FILES_IN_DIR	= 0 -> 1		
	struc {		[MAX_FILENAM MAX_EXTENSIO)N + 1];		(plus space for nul) (plus space for nul)		Fname FileA	nIndexBlock
	} fil	long nInde es[MAX_FILES.	-			dex block is on disk array of these		\0\0	0

Now let's create a file "FileA.txt" inside DirA: Allocate two blocks for FileA, and update root & dirA & FileA metadata

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA	FileA	FileA's initial data block		

```
struct cs1550_index_block
{
      //All the space in the index block can be used for index entries. Each index
      //entry is a data block number.
      long entries[MAX_ENTRIES_IN_INDEX_BLOCK];
      entries
};
```

Now let's create a directory DirA: Save changes to "disk"

Block 0	Block 1	Block 2	Block 3	Block 4	
root	DirA	FileA	FileA's initial data block		

The creation of "/DirA/FileA.txt" modifies data inside block 0, 1, and 2. We don't have to modify the initial data block for fileA. So we need to save these changes (block 0,1,2) into "disk".

Hints

• In each function, you'll open and read metadata from ".disk", remember to save changes and close ".disk" before return

• Example of reading root metadata using fopen/fread

```
struct cs1550_root_directory root_dir;
FILE * fp;
// open .disk for binary read/wrirte
fp = fopen (".disk", "rb+");
// read 1 item whose size is sizeof(struct cs1550_root_directory) to root_dir.
fread (&root_dir, sizeof(struct cs1550_root_directory), 1, fp);
```

You may use fseek to set the position indicator for reading/writing

Hints

- Test commands:
 - Is (with/without –al option), e.g., Is –al testmount
 - mkdir, e.g., mkdir testmount/DirA
 - echo, e.g., echo "" > testmount/DirA/FileA.txt
- Run you program with –d option for debugging
 - You'll need two terminal windows, one for running you program and showing debug messages, the other for running test commands
 - You'll see the triggered syscalls in the debug messages