Operating Systems

HMW2 Report

Arda Çaltepe

Directory entry contents:

name: to hold directory or filename
parent: to hold a parent of directory or filename
firstblock**:** start block of data
directory **:** defines directory or not
size **:** size of the directory or file
permissions **:** read write or read-write rights of the owner.
CreationTime **:**creation of the file
modificationTime**:** modification time of the file
pass **:** password if exists
exists**:** defines deleted or not
haspw**:** directory or file contains a pw or not.

- Directory table just a simple array in runtime then written into FileSystem file after commands are executed.
- Super block are started at 0-1 and used to manage file system information such as number of blocks , free blocks ,fat blocks and directory list blocks.
- Free table holds 0 or 1 depends or free situtation of the blocks. Defines which block is free.
- Fat table : keeps index of next block. Implements how file are stored.
- Directory list : Holds metadata of the files and directories.

- Superblock starts at the first 0-1 index.
- Table of free blocks starts at right after super blocks
- Fat table starts right after Free blocks table like “sizeof(SuperBlock) + number_of_blocks *s izeof(int) bytes from the beginning of the file.
- Directory table starts right after fat table like “sizeof(SuperBlock)+ 2 * (number of blocks * sizeof(int)) bytes from the beginning of the file.

- Handling permissions is established in a way that in write command occurs program copied the permissions of written file into directory metadata then checks if file has necessary permission for write (ex: 0 → read1 , 2→ write/read). Also in read command program checks the file permission in terms of write (ex: 1 → write , 2→ write/read). If they not have permissions program returns from that function immediately.
- For password protection we have a field called pw in directory structure. When addpw command executed it sets pw field as the argument supplied. In default no pw provided. Also in directory structure haspw bit set to 1 to the further analysis. In “chmod” , “delete” and “read” commands are checked before executed in terms of password control.

Dir

Takes parameters as splitted path array , directory entry array and size of the path array. First checks the childs parents exists or not. For example “arda/arda1/arda2” checks arda, arda1 and arda2 is exists or not in directory table. At the end list the all directory tables entries if their parent is targeted directory. Also control the file or directory.

Mkdir

Mkdir function first checks the existence of the path just like in the dir but with a one difference is that last item is not checked. It is checked in terms of not existence, if not exists new directory created as with parent(next item in the splitted path array) of it. With these change directory table updated but also directory entries should be updated in file for this change we are setting a bit as change in directory = 1.

Dumpe2fs

Lists block count , free block , number of files and directories and also lists all occupied blocks. Shows the block size , and number of blocks. Counts the number of free blocks with free table occupation. Iterates through all directory entries, counts the file and directory count based on directory field. List the occupied blocks according to the exits and directory field and shows them one by one which are files by the way. Starting from field startBlock until the -1 shows occupied blocks from files.

Rmdir

Removes the directory and it’s contents. First checks the given path is exists or not then deletes files iterating through all child files in directory. After that erases itself and it’s subdirectories recursively. Just sets the directory table entry with some empty and zero values and sets the directory entry changed flag to 1 as well as change in fat table and change in free table.

Delete

Removes the files with checking existence of given path and the file. If file exists gets the location in fat table with firstBlock field then sets fat table content to -1 until allocated area is finished for that file. Free table is also managed in this way , current location puts into the free table array then sets itself to -1. At the end directory table modified with empty and 0 values for clear the entry. Also fat_table changed is set to 1 as well as directory and free table changed is set to 1 because we have changed their contents we should have to write it to filesystem file.

Write

Reads given file then writes it’s content into blocks. First start with checking the given path is valid also creates a file named as string at the end of the path. Also gets the permissions from the target file and writes into file created in the system. Then reads all content of target file into buffer. Reads this buffer after determining the file block and fat table location then writes the buffer content. At the end copies the necessary info for the directory entry permissions and also updates the parent directory total size and itself. Sets the fat table change and free table change bits to store changes into filesystem.

Read

This function reads the content of the file from block and first checks the given path is valid or not. If the path valid then reads the content of the file blocks and writes the buffer into the new created file. There is no change in fat table or free table that is why we do not need to write results again in our file system.

Chmod

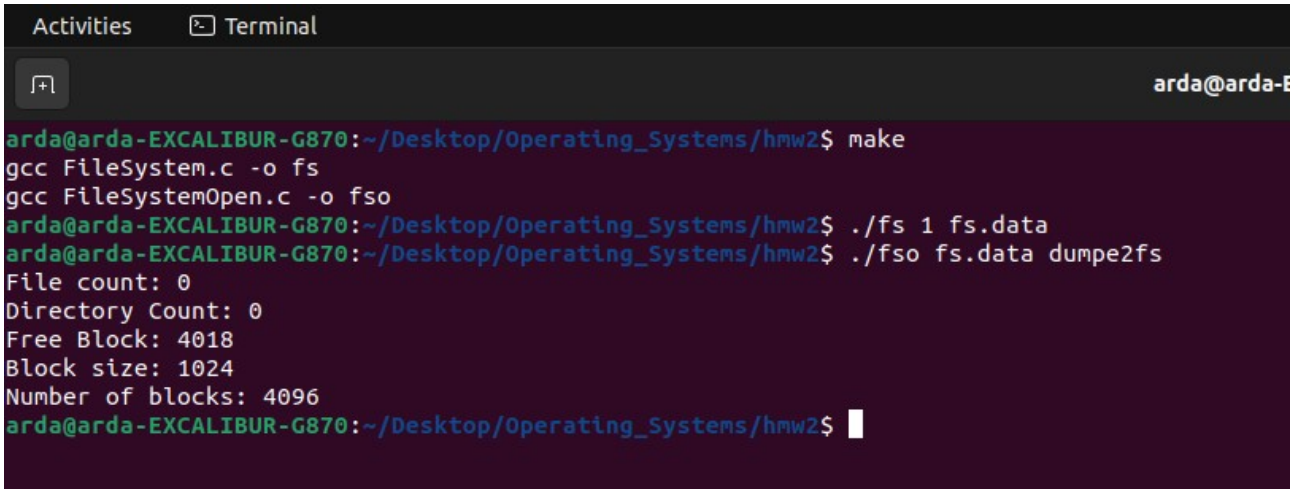
This function changes the mod of the given file path. But first ensures that the if file has password ,it is need to be given. Iterates through directory_entry table array then finds appropriate results then changes it permissions according to the given parameter. If "r" sets as 0 or "w" sets as 1 or "+rw" sets as 2. These permissions will be checked in read write commands execution stage.

Addpw

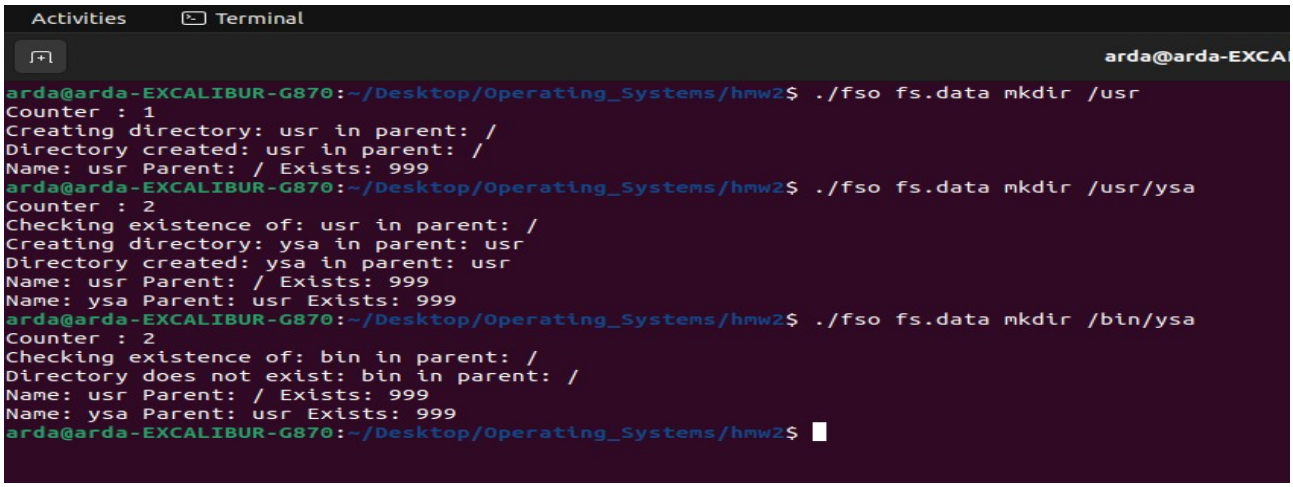
This command provides a password the given path. First checks if that given file is exists then iterates through all files and sets it's password with given parameter. After this set commands like read write chmod will be executed with password.

TESTS

1) Filesystem creation



2) Mkdir



3)Write and Dir

```
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data write usr/ysa/file1 linuxFile
Counter : 3
Counter : 3
Owner can read: yes
Owner can write: yes
Owner can execute: no
Writing to block: 78
Content: linuxFile

arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data write usr/file2 linuxFile
Counter : 2
Counter : 2
Owner can read: yes
Owner can write: yes
Owner can execute: no
Writing to block: 79
Content: linuxFile

arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data write /file3 linuxFile
Counter : 1
Counter : 1
Owner can read: yes
Owner can write: yes
Owner can execute: no
Writing to block: 80
Content: linuxFile

arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data dir / linuxFile
Name                Parent              First Block  Directory  Size       Permissions  Creation Time        Modification Time    Password   Exists
usr                 /                   0            1          10         0            1717854635          1717854635           0          999
file3               /                   80           0          10         2            1717855209          1717855209                      999
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data dir /
Name                Parent              First Block  Directory  Size       Permissions  Creation Time        Modification Time    Password   Exists
usr                 /                   0            1          10         0            1717854635          1717854635           0          999
file3               /                   80           0          10         2            1717855209          1717855209                      999
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$
```

4) Delete and Dumpe2fs

```
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data dumpe2fs
Blocks file2: 79
Blocks file3: 80
File count: 2
Directory Count: 2
Free Block: 4016
Block size: 1024
Number of blocks: 4096
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$
```

5)Read

```
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data read /usr/file2 linuxFile2
Counter : 2
Counter : 2
File: file2
File: file2
Start: 79
Buffer: linuxFile

arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ cmp linuxFile linuxFile2
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$
```

6)Chmod and Read

```
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data chmod /usr/file2 +rw
Counter : 2
Counter : 2
Permissions changed to 2: file2 in parent: usr
arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$ ./fso fs.data read /usr/file2 linuxFile2.data
Counter : 2
Counter : 2
File: file2
File: file2
Start: 79
Buffer: linuxFile

arda@arda-EXCALIBUR-G870:~/Desktop/Operating_Systems/hmw2$
```

7) Addpw and read