Laboration 3

Primary functions

Format

Format is used as the name suggests to format the disk so that it is starting on a clean slate. This was done by setting up the local fat with the corresponding start values as well as setting up the directory array for the root directory (“/”).

Create

The create function takes in a prefered filename from the user and then proceeds to gather data to input into the file. It then goes through the path and checks so the file location can be found and then proceeds to call the “CreateFile” function before exiting the function call.

Cat

The cat function loops through the path if it is not only containing the filename. For each iteration of the search we call the FindFile help function to return the index in the directory array for this folder, and if not found it returns -1. When the correct block of data is discovered the help function “ReadFromFile” is called which returns the data from the file in the form of a string which then is printed.

Ls

ls takes the current block and directory-array as parameters. First prints a row in the terminal with name, type, accessrights and size. Then the directory-array is looped and all directories and files are printed with name, type, accessrights and size to the terminal.

Cp

The cp function loops through the two given paths and retrieves the data from the first path by calling “ReadFromFile” and creates a new file using the “CreateFile” function. The function also prints out error messages if certain parts of the path can’t be found.

Mv

This function loops through the paths and finds the correct block to apply the changes and uses “FindFile”, “ReadFromFile”, “rm” and “CreateFile” to correctly apply changes.

Rm

Rm loops through the path by using “FindFile” as well as “splitStr” help functions and after finding which block the file is in it either removes it immediately (if the path only contains one item) or calls itself with changing the currBlock global variable to the correct and inserting only the file name. When it’s done it changes the currBlock back to the correct number.

Append

Uses the same method of looping through the paths as previous functions and then calls for “rm” for both files and creates a new file using “CreateFile” with data from “ReadFromFile”.

Mkdir

The mkdir function loops through the path like the other functions and then uses the “CreateDir” help function to create the function at the correct block.

Cd

Loops through the given path and switches the current working directory address and current working block in the RAM accordingly using the “FindFile” help function.

Pwd

prints the current working directory using the variable stored in the RAM.

Chmod

Changes the access rights of a given file using the FindFile help function.

Help functions

FindFile

This function takes in the name of a file and a memory block (the block contains directory entries in all the use cases), searches through the directory entries for a file matching the indata. If the function finds a match it returns the index in where the directory entry is in the directory list, if no match was found it returns “-1”.

ReadFromFile

Takes in an dir_entry object to read. The function reads through all corresponding fat table indexes until it faces EOF and then returns a string with the data.

CreateFile

Takes in the desired name of the new file and the desired data to input into the new file. The function creates a new dir_entry object that gets the new attributes. It then adds the object into the first open space of the directory array and writes it into the directory block as well as their own data blocks (adds corresponding entries in the fat table of course).

CreateDir

Creates a new directory with the desired name in the same fashion as CreateFile except this time it only allocates one block of memory for the new object.

SplitStr

Splits the input string by the delimiter “/” and returns a string vector with all the parts of the input string.

Access_helper

Is called from a function that needs to check the access rights. Acces_helper finds the file and compares the given access rights given as a parameter and what access rights are on the file. Returns true if the calling function has permissions to do operations on the file otherwise returns false.

Access_checker

Checks the access rights for a given path. The function takes an integer (access rights) and a filepath as a parameters. Then the function finds the file and checks the accessrights of the file compared to the ones given as parameter. The function returns 1 if is a match 0 if not and -1 if file not found.

isDir

isDir takes a path as a parameter and checks if it is a file or a directory. The function loops through the blocks until it finds the file. Then the function checks the type of the file and if its a directory it returns true otherwise returns false.