CSE 312 MIDTERM FILE SYSTEM DESIGN REPORT

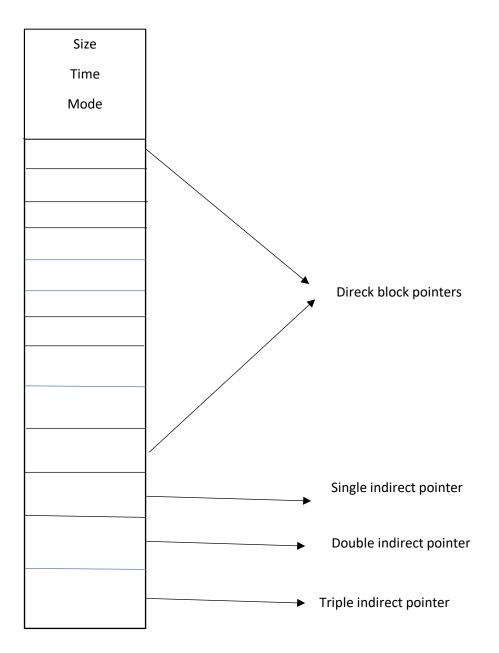
(PART1)

İBRAHİM AKBULUT 151044077

STRUCTERS

super_block: This struct is responsible for holding number of i-nodes in filesystem, size of filesystem's block(int KBs), starting address(block number) of i-nodes, starting address(block number) of data blocks, size of filesystem's inode(in Bytes) and block number of root directory of filesystem.

i_node: This struct is responsible for holding mode of an file, size of an file, time informations of a file and block numbers of data blocks of an file. These structer has 10 direct pointer, 1 undirect single pointer, 1 undirect double pointer, 1 undirect triple pointer.



directory: This structer is responsible for holding i-node number of a file and file name of a file. This struct will be written to entry blocks as an entry of a directory. File name is an array of characters with 14 slot. I-node number is an **short** type variable.

Directory-

i-node number	filename

14 is the i-node number of itselfs

•	14	

Free blocks and inodes are kept in first block after super block in filesystem. There are char bitmaps for both blocks and inodes. Of course these bitmaps have changeable sizes. So, size of free space management in the filesystem is not constant. Example: if file system has 4 kb block size and 400 inode then free space management has 1 block size even more than necessary.

Functions

- -All functions (not bonus functions) were implemented. And all functions work fine with many error situation(can handle with many error).
- -Path names to the program are must be in format "/usr/directory1/directory2"
- -2 / character after each other is invalid path name.
- -There cant be / character in the end of path name.
- -Path name (File name) must be start with a / character.

read_directory_block(): These function reads entries of a directory block. Then return all entries as vector.

search_file_in_directory(): This function checks an i-node has a specific file or not.

search_file_in_block(): This functions checks an directory block has a specific file or not.

find_empty_index_in_directory_block(): Look for a avaible directory place in directory block.

find_place_to_locate_block_index(): Loof for an avaible directory place and return block number of that location and index of empty directory block.

del(): These method take filename (in path) as argument. Other arguments are provided by filesystems.data file. If file that is deleted is a directory then an error occurs. If file is not in filesystem then error occurs. If a file is deleted then parent directory's modification time changes. Del() can not delete a directory.

mkdir(): If file name is name of existing directory then an error occurs. If path name has an file over actual file then an error occurs that tell "This is not a directory". If number of avaible inodes and blocks are not enough to create a directory then an error occurs.

write(): If given Linux file is not exist then an error occurs.if given file name to write Linux file to it is name of a directory then an error occurs. If given file name is name of an existing file then first delete that existing file then create this file with Linux file(overwriting).

list(): This function list all files under given path.

read(): If given file is not exist in filesystem then an error occurs.

rmdir(): rmdir can not delete a file. It can delete only directories.

dumpe2fs(): This list all superblock informations of filesystem. Plus, it lists total number of blocks, free inodes count, free blocks count, used inodes count, used blocks count. Plus, it lists used inodes with information of which data blocks used by that i-node, what is name of that i-node, what is the mode if i-node(d or f).