CSE 312 – Operating Systems Midterm Exam Project Report – 171044079 Gökçe Nur Erer

Directory Structure and Directory Entries

In this file system, directories are implemented with a directory block. In a directory block, there are some attributes along side with the directory entries:

- Maximum entry count that can be added to the directory block,
- Current entry count of the directory block,
- A variable that checks if the directory block is full,
- A variable that checks if the directory block is free.

In a directory block besides these variables, a collection of directory entries are held. The structure of the directory entry is as follows:

Directory/File Name	Number of the I-node that corresponds to the file/directory

Free Blocks and Free I-nodes

In this file system, blocks and i-nodes are grouped. For example single indirect blocks have their own group where there are both free and occupied single indirect blocks. I-nodes also have their own group where there are both free and occupied i-nodes. To use a free block or i-node the only requirement is to pass through the requested block's group and find the first available block.

In this file system, blocks that hold data for directories and files are seperated. Blocks that keep data about files are called data blocks and blocks that keep directory entries of directories are called directory blocks. They have the same goal but in directory blocks, directory entries are held instead of normal byte data.

I-Node Structure

Attributes
Block Pointer 1
Block Pointer 2
Single Indirect Block Pointer
Double Indirect Block Pointer
Triple Indirect Block Pointer

Attributes include the following:

- I-node type: the type corresponding to the i-node. Can be file, directory or free (a not assigned i-node)
- I-node number
- Link count
- A variable that represents if the i-node is free or not
- The size of the file/directory
- Last modification of the file/directory
- Name of the file/directory

Superblock

In this file system, superblock contains variables that control the file system flow. These variables are:

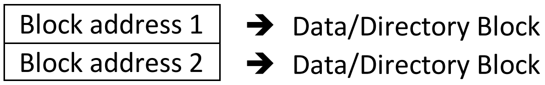
- Magic Number: File system uses this number to check if the given disk file has the correct format.
- Block Size: Size of blocks in the file system, given by user.
- Maximum I-node Count: Depending on the sizes and counts given by the user sometimes it is not possible to create as many i-nodes as the user requested. In that case file system uses this value to create as much i-nodes as it can.
- Maximum File Count: Depending on the i-node count, 80% of the i-nodes are reserved for files. This value shows how many files can be created on the system.
- Maximum Directory Count: Depending on the i-node count, 20% of the i-nodes are reserved for directories. This value shows how many directories can be created on the system.
- Maximum Single Indirect Block Count: For each i-node 7 single indirect blocks are needed (which will be explained below), depending on the i-node count of the system this value is calculated so that many single indirect block can be allocated.
- Maximum Double Indirect Block Count: For each i-node 3 double indirect blocks are needed (which will be explained below), depending on the i-node count of the system this value is calculated so that many double indirect block can be allocated.
- Maximum Triple Indirect Block Count: For each i-node 1 triple indirect block is needed (which will be explained below), depending on the i-node count of the system this value is calculated so that many triple indirect block can be allocated.
- Maximum Directory Block Count: For each i-node that holds a directory, 16 directory blocks are needed (which will be explained below), depending on the i-node count of the system, this value is calculated so that many directory blocks can be allocated.
- Maximum Data Block Count: For each i-node that holds a file, 16 data blocks are needed (which will be explained below), depending on the i-node count of the system, this value is calculated so that many data blocks can be allocated.
- Maximum File Size: Since the data block count is constrained for each file, the maximum file size is the total of all data blocks an i-node holds.
- Maximum Directory Size: Since the directory block count is constrained for each directory, the maximum directory size is the total of all directory blocks and i-node holds.
- I-nodes Starting Position: Address where i-nodes start.
- Directory Blocks Starting Position: Address where directory blocks start.
- Data Blocks Starting Position: Address where the data blocks start.
- Single Indirect Blocks Starting Position: Address where single indirect blocks start.
- Double Indirect Blocks Starting Position: Address where double indirect blocks start.
- Triple Indirect Blocks Starting Position: Address where triple indirect blocks start.
- Free Data Block Count
- Free Directory Block Count
- Free Single Indirect Block Count
- Free Double Indirect Block Count

- Free Triple Indirect Block Count
- Free I-node Count
- The current file count of the system
- The current directory count of the system
- The maximum directory entry count that a directory block can hold : This value depends on the block size so it has to be calculated in the superblock before creating any directory blocks.

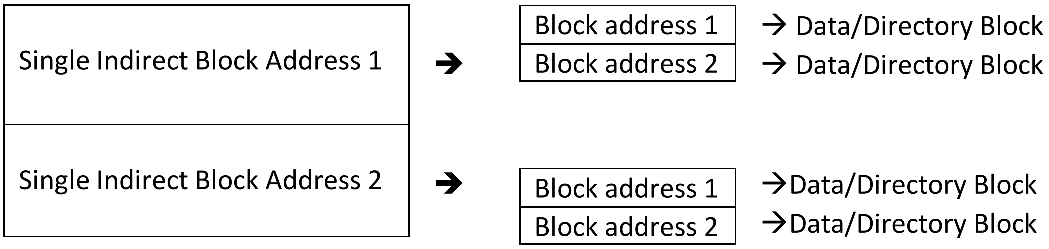
Single, Double, Triple Block Structure

In this file system single, double and triple indirect blocks are structured like this:

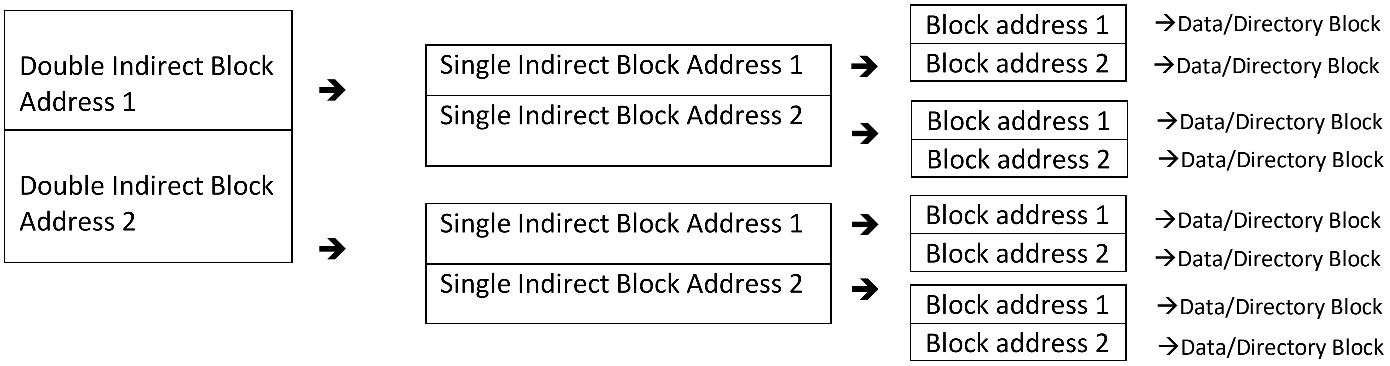
Single Indirect Block



Double Indirect Block



Triple Indirect Block



Each i-node contains 2 direct block address, 1 single indirect block address, 1 double indirect block address and 1 triple indirect block address. This file system's indirect blocks hold 2 address values. Which results:

- 1 single indirect block from i-node
  + 2 single indirect block from double indirect block

+ 4 single indirect block from triple indirect block
= 7 single indirect block for an i-node.

- 1 double indirect block from i-node
  + 2 double indirect block from triple indirect block
  = 3 double indirect block for an i-node.

- 1 triple indirect block from i node

- 8 data/directory blocks from triple indirect block
  + 4 data/directory blocks from double indirect block
  + 2 data/directory blocks from single indirect block
  + 2 direct data/directory blocks from i-node
  = 16 data/directory blocks for an i-node.

Table for the function names in source code corresponding to the operations in part 3:

Function in the source code	**Operation**
In File System class → dumpe2fs function	dumpe2fs operation
In File System class → write function	write operation
In File System class → read function	read operation
In File System class → list function	list operation
In File System class → mkdir function	mkdir operation
In File System class → rmdir function	rmdir operation
In File System class → del function	del operation