Josh Ralls and Darius Thomas

CSC 400

3 December 2019

Project Documentation

**Ext2Reader Class**

The first class that is used is the ext2Reader class which essentially serves as the menu that will allow the user to be able to choose what actions they want to take. First, it gives the users the option to tell what path the virtual disk is in. By using RandomAccesFille the program can open the virtual disk if the user has entered a valid path. If not, they will be told that the path that they have entered invalid. The program will then display options for the user which are a help list, change directory, list directory contents, copy the contents to a host drive, end the program, and finally if the program does not recognize the user's command then it will tell them so.

*Help()*

If the user chooses this option, the user will be met with a print out of the commands that the user is able to use on the menu. The command “CD” allows for the user to change directories, “LS” allows the user to list the contents in the directory, “CP” allows the user to copy a file, and “H” allows the user to display the list of commands.

*ChangeFolder()*

If the user chooses the “CD” command this method will allow the user to change what directory they are currently in. By using a for loop that goes up to the number of folders in the filesystem the user will then be able to see what folders are currently in the file system. The user will then be able to see and select what folder they want to use. If the directory is ever changed it will be saved in the “currentDirPath” variable. If there is ever an error the user will be prompted to what the error.

Parameters:

It- Current inode table.

*ListContents()*

If the user chooses the “LS” command they will be able to see what folders followed by the available files are in this directory.

Parameters:

It- Current inode table.

*CopyFile()*

If the user chooses the “CP” command they will be able to copy a file from the ext2 to the host drive. After selecting this command, they will be asked what file they are wanting to copy and where on the host drive would they like the copied file to be placed in. This method will then use the input of the user to create a folder called “CopiedContents” unto the Host Drive and will then copy the file from the ext2 and place it into the “CopiedContents” folder. String desired file path, String filename, String pathforCopiedContents

Parameters:

desiredFilePath- Path that leads to where the desires file is on the EXT2.

filename-Name of the desired file followed by file type suffix.

pathforCopiedContents-Path to where the file the desired file will be placed on host Drive.

**CommonFunctions Class**

This class will be used to store methods that will frequently come up in traversing through the EXT file system. This class is compromised of the findByteInodeOffset method and the searchBlock method.

*findByteInodeOffset()*

This method is used to find the byte offset that will be needed to properly traverse through an inode object. By using information from the Group Descriptor, Superblock, and inode table this method will return the start address for and inode table as seen in the InodeTable methods of the InodeTable class. To read and translate the information that is in the file the RandomAccessFile class was imported to be able to read the needed bytes in the EXT2.

Parameters:

Sup- Reference created to access the SuperBlock class

Gd- Reference created to access the GroupDescriptor class

inodeTable- Location of the inode table

*searchBlock()*

This method is used in finding the correct value of a descriptor for either a Block Group, Directories, Inodes, or a Super Block for an ext2. For example, if the 32bit Value of the s\_inodees\_count is needed this method will find the correct position of the descriptor and read the needed bytes to return a 32bit integer value.

Parameters:

Seekoffsett-How far from the beginning of the file.

byteOffsett-How far from the seek Offset should the cursor be placed.

length-How many bytes should be read.

raf- Reference created to access the RandomAccess class.

**GroupDescriptor class**

This class is used to find the needed descriptions for the Block Group Descriptor Tables. Variables bloc\_bitmap, inode\_bitmap, Inode\_table, free\_blocks\_count, and used\_dirs\_count will represent the descriptors that are common in the Block Group Descriptor Table. In this class, a GroupDescriptor method and a forBlockGroupDescrip will be used to find the needed integer values.

*GroupDescriptor()*

This method will be used to not only start the forBlockGroupDescrip() method and to catch any IOException that may occur.

Parameters:

raf- Reference created to access the RandomAccess class.

*forBlockGroupDescrip()*

In order to successfully access the bytes that contain the needed properties, another method called searchBlock() from the CommonFunction class will be needed. In the forBlockGroupDescrip() the variables stated before will be returned a matching value after using the searchBlock() method. The parameters used in the searchBlock() are associated with the offset, size, and seekOffsett that is found in nongnu, en.wikipedia, and Science.unitn. These variables will be used when needed in other classes that require information about the Block Group Descriptor Table.

**Inode class**

This class is used to find the needed descriptions for the Inodes of the EXT2. Variables uid, size, atime, ctime, mtime, dtime, gid, links\_count, blocks, and finally flags will represent the descriptors that are common in Inode. In this class, an inode method and a forInode will be used to find the needed integer values.

*inode()*

This method will be used to not only start the forBlockGroupDescrip() method and to catch any IOException that may occur.

Parameters:

raf- Reference created to access the RandomAccess class.

*forinode()*

In order to successfully access the bytes that contain the needed properties, another method called searchBlock () from the CommonFunction class will be needed. In the forInode() the variables stated before will be returned a matching value after using the searchBlock() method. The parameters used in the searchBlock() are associated with the offset, size, and seek Offsett that are found in nongnu, en.wikipedia, and Science.unitn. These variables will be used when needed in other classes that require information about the inode.

*inode()*

This method will be used to not only start the forInode() method and to catch any IOException that may occur.

Parameters:

raf- Reference created to access the RandomAccess class.

*forinode()*

In order to successfully access the bytes that contain the needed properties, another method called searchBlock () from the CommonFunction class will be needed. In the forInode() the variables stated before will be returned a matching value after using the searchBlock() method. The parameters used in the searchBlock() are associated with the offset, size, and seek Offsett that are found in nongnu, en.wikipedia, and Science.unitn. These variables will be used when needed in other classes that require information about the inode.

**SuperBlock class**

This class is used to find the needed descriptions for the Inodes of the EXT2. The integer variables inodes\_count, block\_count, log\_block\_size, log\_frag\_size, blocks\_per\_group, frags\_per\_group, inodes\_per\_group, first\_ino, and inode\_size will represent the descriptors that are common in Inode. In this class, an inode method and a forInode will be used to find the needed integer values.

*SuperBlock()*

This method will be used to not only start the forSuperBlock() method and to catch any IOException that may occur.

Parameters:

raf- Reference created to access the RandomAccess class.

*forSuperBlock()*

In order to successfully access the bytes that contain the needed properties, another method called searchBlock () from the CommonFunction class will be needed. In the forSuperblock() the variables stated before will be returned a matching value after using the searchBlock() method. The parameters used in the searchBlock() are associated with the offset, size, and seek Offsett that are found in nongnu, en.wikipedia, and Science.unitn. These variables will be used when needed in other classes that require information about the SuperBlock.

*Superblock()*

This method will be used to not only start the forSuperBlock() method and to catch any IOException that may occur.

Parameters:

raf- Reference created to access the RandomAccess class.

*forSuperBlock()*

In order to successfully access the bytes that contain the needed properties, another method called searchBlock () from the CommonFunction class will be needed. In the forSuperblock() the variables stated before will be returned a matching value after using the searchBlock() method. The parameters used in the searchBlock() are associated with the offset, size, and seek Offsett that are found in nongnu, en.wikipedia, and Science.unitn. These variables will be used when needed in other classes that require information about the SuperBlock.

**InodeTable Class**

This class is used to find the needed descriptions for the LinkedList directory and the Inode Table of the EXT2. Variables inode, re\_len, name\_len, file\_type, and name will represent the descriptors that are common for the Linked List Directory. In this class reverse bytes, inode, getNumBytes, and Inode Table will be used.

*reversebytes()*

This method will read the inputs and interpret them in the little-endian format.

Parameters:

Len-number of bytes that need to be read.

Raf- reference created to access the RandomAccessFile.

*InodeTable()*

Reads the inode table of the EXT2 and handles going through the folders of the directory correctly

Parameters:

Raf- reference created to access the RandomAccessFile.

Sup- Reference created to access the SuperBlock class.

Gd- Reference created to access the GroupDescriptor class.

inodeTable- Location of the inode table.

*inode()*

Reads the folders or inodes that are found in the inode table.

Parameters:

Len-number of bytes that need to be read.

inodeTable- Location of the inode table.

*getNumBytes*

Gets the bytes that are needed.

Parameters:

Len-number of bytes that need to be read.

Raf- reference created to access the RandomAccessFile.