Joe Tursi

Dr. Rivas

CMPT 220

May 5, 2017

Final Project: File Compression

Abstract:

The following paper is focuses on the semester project of File compression, which is used to reduce the size of the file and to get rid of any redundancy which in return makes the program more efficient. This paper will focus on the purpose of file compression and go into depth of how users will be able to interact with it. The next step will explain what is required from the system in order to execute the project in terms of resources. Moving on, it will be explained what other programs exists for file compression. File compression is very common in there are many different ways to make files more efficient in downloading speed. After explaining the various options to my project, the user will be given guidelines and instructions to run the program properly in order for it to perform the task it was created to do.

Introduction:

The motivation behind this work is to challenge myself as a developer. File compressors are a very common thing and easily accessible throughout the web. I personally have a lot of files on my desktop and it would be cool to run this program and see if it could work on certain text files I have. This paper will continue by giving a detailed system description, by informing the user what the program does and how the user will be able to interact with it. The steps behind this will

be fairly simple and be user friendly to all developers. The requirement of the user will be outlined in order for the user to be able to run the program effectively and successfully without any errors. After that, there will be a brief excerpt explaining the other programs that are available to use for the same purpose, and argue how mine is different. There will be a set of precise instructions that will allow any developer or person be able to clearly understand how the file compression program can be executed.

Detailed System Description:

This program performs a File compression, which is performed by using my code to shrink the size of the data. So the method compress has two arguments, source and destination, which is the file it's going to compress, and where the file is going to be compressed. This method throws the IOException. GZIPInputstream will read a stream of bytes from the FileInputstream and compress it then it will use fileoutputstream to write the compressed byte to a file. The variable read is used in the while loop, reading the bytes from the source file. The bytes are read from the source file using GZIPInputStream and the bytes that are read will be stored in the variable read. The read method returns the total number of bytes into the buffer, or it returns negative one if there is nothing else to read. While the method returns something different from negative one it means they bytes weren't read, and are written to FileOutputStream. We want to write bytes from the buffer, with an offset of zero, with the length of the read variable. After that I closed the FileInput, FileOutput, GZIP Output streams. The program now runs the main method, and in that are two file objects. In those file objects there is the source file path, and the destination file path. Now we use the File Compress method and

made a try and catch block, to catch the IOException, which is then printed out. In the try block I have a compress method written, with the source and destination file arguments. So now the User can run the program and compress the file in java.

Requirements:

A system resource is a resource that is either built-into the system, or kept by the implementation, for example a local file system. So basically it's any physical or virtual competent that doesn't last forever within your computer. The system resources you will need to run the program will be CPU time, memory space, disks, input and output devices, which be allocated by your operating system, allowing the program to run. One of these system resources that you will require in order to run this system is Random access memory or RAM. The file compressor isn't the most effective so I'm guessing it's going to use a lot more RAM than it should. The more RAM that you have on your computer the more programs you can run on your computer. Since this program is reading the file, than writing it into a new one it is going to impact the performance of RAM on your system. I think two gigabytes of RAM will be more than enough to suffice, not causing any visual problems or signs of slowing down the system. Another resource your system will need to run this program, is a java platform in order to be able to run the program.There a number of platforms that are made for different systems which allow the user to run the program. Your system will also require CPU, or the central processing unit, which will carry out the instructions that my program gives it through input and output operations. Inputs is the data received from the program, and the output is the data that is sent

out. So in this case the program is telling the computer to read in a file, and compressing is the output.

Literature Survey:

There exists countless different file compression systems in different languages that should allow you to extract all archives and files, such as ZIP, RAR, ISO, and TAR, and a bunch of other formats. Mine is probably the most efficient and slowest since its run through java. But not all file compressors can be ran on Linux like mine can. My program as its is written can only compress at a slow rate, since its compressing 1024 bytes which is equivalent to just over one kilobyte. My code is different from other in that it is relatively compact, it will carry out the same purpose as others by reducing the size of a file in order to save space and make it more efficient when transferring it over the internet or wherever its sent.

User Manual:

The File Compression program can be executed correctly by putting the path of the file that the user wishes to compress. In the main method there is class name called File, which has a name of an object called source. The source file is the file that you want to compress, a new object was created which calls the compressed file you want to compress. In new file put the destination of the file you wish to have compress. The same thing is written in the code for the destination of the newly compressed file. The class name file is declared with the object destination which is made into a new file. In my code it's called “BrandSpankingNew.rdf” which is where the compressed “wiki.rdf” file will be compressed to. After you write in the path of the

file you wish to compress, compile the program and then run it. The file should be compressed in the new location, but you won't be able to read it because it wasn't decompressed, which isn't the purpose of this project.

Conclusion:

In this paper we explored the concept of file compression and then applied it by making a program that carries out that purpose. We described what the system does with the program and gave instructions on how users would interact with them. We then walked through the system resources that would be needed in order to run this or any program in general. We then did research on other file compression projects that differ from the one that was programmed, and how they vary from the one written in this project in efficiency.

In conclusion, the program runs successfully and the objective of the project was met. I learned how file compression is used to reduce the file size, which return takes up less disk space of the user's computer and is more efficient to transfer. The significance of this project was learning all the positives things that come through file compression and how it could be used in future projects. File compression reduces the space needed to store data, which gives the user more valuable space on their hard driver. In this particular program I focused on files such as text or for mac a rdf file. Compression will get rid of any redundant patterns and replace them with identifiers.

References:

Liang, Y. Daniel (2014-01-03). Introduction to Java Programming, Comprehensive Version (Page 2). Pearson Education. Kindle Edition.