**1. Implementation**:

COSC 5330-01  
Multimedia Processing  
  
INSTRUCTOR:  
Prof. Jiangjiang (Jane) Liu  
  
  
Implement encoder and decoder for

LZW compression algorithms  
  
  
Bhumit Patel  
L20382154  
11/16/2015

This project deals with the use of lzw algorithm to encode and decode files. I have taken several files that includes text file, image file, audio file, dll file and exe file. I encode all these files by using LZW implementation and noted compression ratio. I used winzip as standard tool for compression and noted compression ratio to compare with my LZW implementation.

When I compress the above files using my source code I get a compressed file with format lzw. The compressed files with extension lzw are taken and they are decompressed.

* 1. **Encoding**

The input files of text, image, audio, dll and exe files are taken and compressed using our source code.

**Result**

The files are compressed and the compressed files with lzw format are created. The compression ratios are noted for the compressed files.

* 1. **Decoding**

The compressed files with format lzw are taken and they are decompressed using the source code Decoding.java.

**Result**

The files are decompressed and original files are created.

* 1. **Winzip Encoding**

Now the files are compressed using the winzip encoder and the compression ratios are noted.

**2. Comparison**

I compare the compression ratios of winzip and my lzw Implementation. I find that the compression ratios evolved by using lzw algorithm are better than compression ratios evolved using winzip. Thus lzw algorithm is the best compressing technique.

**3. Methodology**

I am using LZW encoding algorithm to compress the files and LZW decoding algorithms to decompress the compressed file.

**3.1 LZW Encoding Algorithm**

* If the message to be encoded consists of only one character, LZW outputs the code for this character; otherwise it inserts two- or multi-character, overlapping (The last character of a pattern is the first character of the next pattern) distinct patterns of the message to be encoded in a Dictionary.
* The patterns are of the form: C0C1 . . . Cn-1Cn. The prefix of a pattern consists of all the pattern characters except the last: C0C1 . . . Cn-1
* Pseudo-Code

Initialize Dictionary with 256 single character strings and their corresponding ASCII codes;

***Prefix*** *← first input character;*

***CodeWord*** *← 256;*

*while(not end of character stream)*

*{*

***Char*** *← next input character;*

*if(****Prefix + Char*** *exists in the Dictionary)*

*Prefix ←* ***Prefix + Char****;*

*Else*

*{*

***Output:*** *the code for* ***Prefix****;*

*insertInDictionary( (CodeWord ,* ***Prefix + Char****)*

*CodeWord++;*

***Prefix*** *←* ***Char****;*

*}*

*}*

* 1. **LZW Decoding Algorithm**
* The LZW decompressor creates the same string table during decompression.
* Initialize Dictionary with 256 ASCII codes and corresponding single character **strings** as their translations;
* Pseudo Code

*PreviousCodeWord ← first input code;*

***Output:*** *string(PreviousCodeWord) ;*

*Char ← character(first input code);*

***CodeWord*** *← 256;*

*while(not end of code stream){*

*CurrentCodeWord ← next input code ;*

*if(****CurrentCodeWord*** *exists in the Dictionary)*

*String ← string(CurrentCodeWord) ;*

*else*

*String ← string(PreviousCodeWord) + Char ;*

***Output:*** *String;*

*Char ← first character of String ;*

*insertInDictionary( (****CodeWord*** *, string(****PreviousCodeWord****)* ***+ Char****) );*

*PreviousCodeWord ← CurrentCodeWord ;*

*CodeWord++ ;*

**4. Testing Results:**

Here I can compress and decompress 5 different type of files (i.e. audio files, EXE files, image files, data files-with .doc and dll files).I develop this application using JAVA, and I used HASH TABLES in JAVA to build both compression and decompression techniques.

Word File- Sample word file.doc

Image File- Sample image file.jpg

Audio File- Sample music file.mp3

DLL File- Sample dll file.dll

EXE File- Sample exe file.exe

Thus the files are compressed and the size is reduced.

After compression the files I have all the compressed files ending with **.lzw**

**Compression Analysis:**

In Following Table, we have calculate compression ratio using following formula:

LZW Compression Ratio = Original Size/LZW Size

WINZIP Compression Ratio = Original Size/WINZIP Size

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **File Name** | **Original Size** | **LZW-SIZE** | **LZW Compression Ratio** | **WINZIP SIZE** | **WINZIP Compression**  **Ratio** |
| Sample word file.doc | 1080KB | 1043KB | 1.04 | 375KB | 2.88 |
| Sample music file.mp3 | 2640KB | 3613KB | 0.73 | 2538KB | 1.04 |
| Sample exe file.exe | 10116KB | 8184KB | 1.24 | 1918KB | 5.27 |
| Sample dll file.dll | 3640KB | 3546KB | 1.03 | 1349KB | 2.70 |
| Sample image file.jpg | 1069KB | 1511KB | 0.71 | 1058KB | 1.01 |

The above table shows the compression ratios of all the files compressed using LZW and WINZIP.

**Compression Ratio**

**5. User Manual:**

**User Manual:**

Here, I have taken following file for compression and decompression:

Word File- Sample word file.doc

Image File- Sample image file.jpg

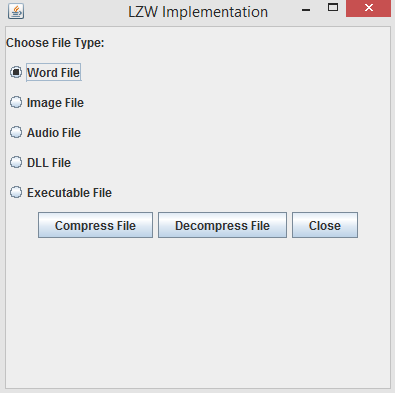
Audio File- Sample music file.mp3

DLL File- Sample dll file.dll

EXE File- Sample exe file.exe

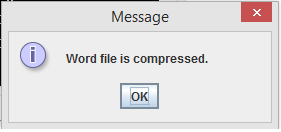
User have to select radio button for compression any of the following types:

Word File, Image File, Audio File, DLL File or Exe File. (**Refer Following Screenshot**)



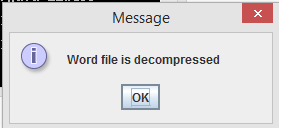
1. To Compress Files:

After selecting any radio button, user can click on compress button to compress files. After compressing file, Following Message box is displayed:



1. To Decompress Files:

To decompress file, user have to select any radio button and click on decompress button. If File has already encoded then decompression will be done successfully and message box will be displayed as under:



If File is not encoded earlier then no file is found for decompression. So following message will be displayed as under: 