MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

"KHARKIV POLYTECHNICAL INSTITUTE"

Department of Computer Engineering and Programming

«Software Means of Information Protection »

*Laboratory work report No 4*

*Topic: «* *Data compression»*

Student:

Group. KH 919 i.e.

Okechukwu Chukwuemeka Onyekwere-Dike

Verified by:

Lecturer Viktor CHELAK

Kharkiv – 2022

***Purpose of work:***

Development of a program for compressing information and returning to the original state without losses.

***Individual task:***

* Information compression program (archiving). As information, use a copy of the file with the developed program.
* Program to return the archive file to its original form (unzip).

Variant 8: 2 - huffm-st

***Algorithm of the program:***

*#Compressor program:*

The program uses Huffman coding for compression of files. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol.

For creating a Huffman tree for specific data:

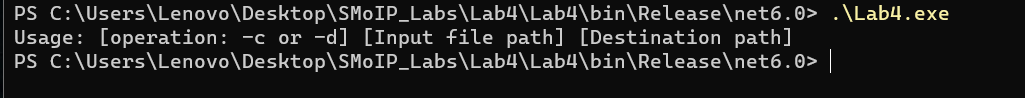
1. Create a node for each character with its frequency
2. Arrange list of nodes ascending frequency.
3. Combine first two nodes and join them with new parent node, with a frequency of the combined nodes.
4. Insert this new node into the ordered list.
5. Repeat 4) and 3) until we get a single node which is the root.
6. Convert input data (file) into new codes (Huffman codes).
7. Convert encoded string to hexadecimal.
8. Serialize the root of tree, encoded string (hex format) and the length of the encoded string in binary format into a bin file.

*#Decompressor program:*

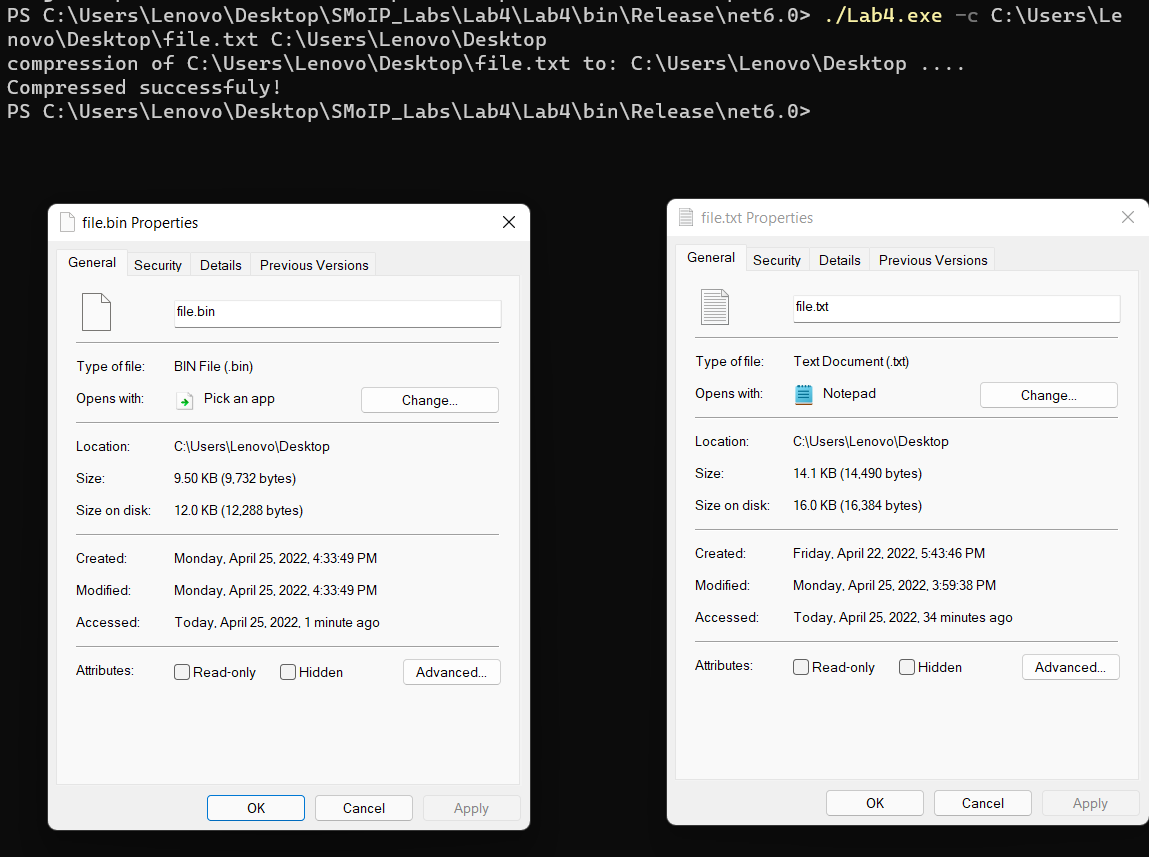
For this part pf program we deserialize data from binary file, we get the root, encoded data in hex format and the length of its binary format.

First, we convert encoded data to binary format, and we use for each loop and a condition that breaks the loop after reaching a number of loops that equals length. And for each character in loop that represents a bit if the bit is ‘1’ we go to the left of the current node and if it’s ‘0’ we go to the right until we reach a final node with value and without any children nodes.

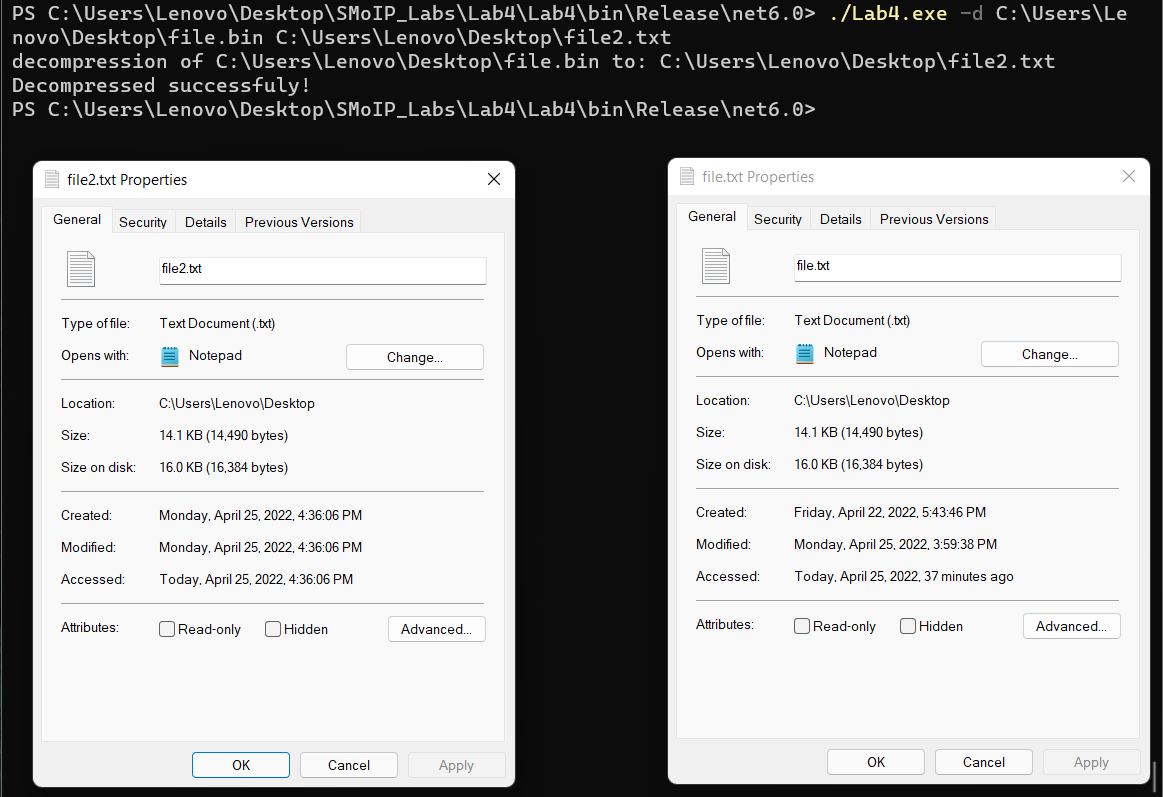
***Screenshots of the program***:



*Figure 1 – Usage of the program*

**

*Figure 2 – Compression of text file (file.txt)*

**

*Figure 3 – Decompression of the file into file2.txt*

**Source Code:**

[*https://github.com/Elh-Ayoub/SMoIP\_Labs/tree/main/Lab4*](https://github.com/Elh-Ayoub/SMoIP_Labs/tree/main/Lab4)

**Conclusions:**

For this laboratory work, I have gained principles of developing a program for compressing data in files, and return it to original state without any losses. Also serialize, deserialize data (multi types) into and from binary files.