**“Implementing Compression Algorithm and Secure Transmission over UDP Protocol”**

***A***

***Project Report***

*submitted in partial fulfillment of the*

*requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Specialization in**

**OSOS**

**by**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Harsh N Mathur** | **R100218020** |
| **Harshit Chauhan** | **R100218021** |
| **Pranay Mahajan** | **R100218037** |
|  |  |

***under the guidance of***

**Mr. Ankit Vishnoi,**

Assistant Professor,

Department of Systematics

School of Computer Science

****

**Department of Cybernetics**

**School of Computer Science**

**University of Petroleum & Energy Studies**

**Bidholi, Via Prem Nagar, Dehradun, UK**

**Aug-Dec, 2020**

upes-new-logo

**CANDIDATE’S DECLARATION**

I/We hereby certify that the project work entitled “Implementing Compression Algorithm and Secure Transmission over UDP Protocol” in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Open Source And Open Standard and submitted to the Department of Cybernetics at School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August**, **2020** to **November**, **2020** under the supervision of **Mr. Ankit Vishnoi, Assistant Professor, Department of Systematics.**

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

**Harsh N Mathur Harshit Chauhan Pranay Mahajan**

**R100218020 R100218021 R100218037**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 20th November, 2020 **Mr. Ankit Vishnoi**

Project Guide

**Dr. Monit Kapoor**

Head –Department of Cybernetics

School of Computer Science

University of Petroleum & Energy Studies

Dehradun – 248001 (Uttarakhand)

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Mr. Ankit Vishnoi**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of the Department, **Dr. Monit Kapoor**, for his great support in doing our project in **Area** at **SoCS**.

We are also grateful to **Dr. Manish Prateek Professor and Dean SoCS**, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Harsh N Mathur** | **Harshit Chauhan** | **Pranay Mahajan** |  |
| **Roll No.** | **R100218020** | **R100218021** | **R100218037** |  |

**ABSTRACT**

Over the last few decades there has been a tremendous increase in the amount of digital data produced and transmitted, representing text, images, video, sound, computer programs, etc. With this trend expected to continue, it makes sense to pursue research on compression algorithms.

Compression algorithms reduce the repetition of data without actual loss of data.

The algorithm proposed here deals with compression of text files using word compression techniques by first Huffman coding the input file channel encoding the source coded file and transmitting over UDP protocol.

With its fast-compressed feature, Word Based Compression Algorithm is a great solution, especially for transmission of data that is large and it helps in compression without the loss of data.

It helps in reduction of both compression ratio that is defined as the ratio between the uncompressed size and compressed size and transmission time of data over a channel.

**TABLE OF CONTENTS**

**S. No. Contents Page No**

1. **Introduction 7**
   1. Requirement Analysis 8

1.1.1. Software Requirements 8

1.1.2. Hardware Requirements 8

* 1. Main Objective 8
  2. Pert Chart 9

1. **System Analysis 9**

2.1. Existing System 9

2.2. Motivations 10

2.3. Modules10

2.3.1. Datacompression and decompression 10

using Huffman Encoding

2.3.2. Socket programming 10

1. **Design 11**
   1. Designing and Development 11
      1. Methodology 11
      2. Flowchart 12
2. **Implementation 13**
   1. Algorithm 14
3. **Output Screens 16**
4. **Limitations and Future Enhancements 18**
   1. Limitations 18
   2. Future Enhancements 18
5. **Conclusion 18**

**References 19**

**LIST OF FIGURES**

**S.No. Figure Page No**

1. Pert Chart 8
2. Flow Chart for flow of system 12
3. Huffman Encoding flowchart 14
4. Socket programming flowchart 15
5. Output Screens 16

1. **INTRODUCTION**

With the advancement of technology, the amount of data produced has been enormously increasing but the storage being provided is not sufficient to meet the increasing demands. With the help of Word-Based compression algorithm, we can efficiently compress text files for better storage. This can not only be used to reduce storage space requirement but also reduce input/output load.

Data compression is a technique by which the same amount of data is transmitted by using a smaller number of bits. It is the process of modifying, encoding or converting the bits structure of data in such a way that it consumes less space on disk. The compression works in the following way first the message is compressed using encoding algorithm and then it is decompressed using decoding algorithms to get the desired data.

One of the most important algorithm to implement compression is that of Huffman Encoding and it is used for lossless compression of data. In this it uses variable length encoding where length codes are assigned on the basis of frequency of characters. The character which has the maximum frequency is allotted smallest code and character which has least occurrence gets the highest code.

We compress the file and send it to the receiver the desired file and it is all done with the help of socket programming. Socket programming is a way of connecting two nodes on a network to communicate with each other

**1.1 Requirement Analysis**

**1.1.1 Software Requirements:**

* Operating System : Windows 10/8/7 (32- or 64-bit) / Linux
* Software : Text Editor
* Compiler : GCC

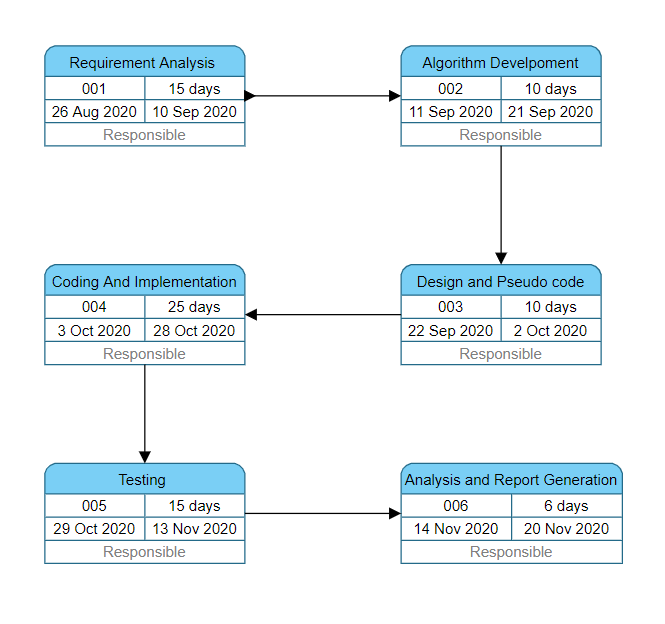
**1.1.2 Hardware Requirements:**

* Computer System : Two –(32- or-64-bit) Core i5 or higher
* Processor : Dual Core or Higher
* RAM : 512 MB or Higher
* Disk Space : 512 MB

**1.2 Main Objective**

* To establish a connection between CLIENT and SERVER using socket programming.
* To implement compression algorithm to compress the text file.
* To implement de-compression algorithm to de-compress the text file.
* To calculate the compression ratio, and space saving percentage.

**1.3 Pert Chart Legend**



# **2. System Analysis**

**2.1. Existing System**

There are many softwares available in the market which provides the compression and decompression of text files. For example, WinZip is a trialware file archiver and compressor.  It can create archives in zip file format, and unpack some other archive file formats.

WinZip stores the compressed file in .zip format but sometimes there is a problem of data loss during decompression of file which results in lossy compression and also sometime people are not able to open the file but our project guarantees 100% lossless compression and decompression of file.

**2.2. Motivation**

There has been unprecedented growth in amount of data. Over the years the amount of data required has increased exponentially. As amount of data is increased, the amount of data to be transferred is also increased and so as the time taken and memory required. So we need to compress the data to decrease the transmission time.

**2.3. Modules**

**2.3.1. Data compression and decompression using Huffman Encoding**

Huffman encoding is used for lossless compression of data. Huffman, uses variable length encoding where length codes are assigned to characters on the basis of frequency of occurrence of character. The character which has the maximum frequency is allotted with smallest bit code and character which has least occurrence gets the highest bit code.

**2.3.2. Socket programming**

Socket programming is a way of connecting two nodes on a network to communicate with each other. In this the compressed file generated through huffman is transferred to the desired receiver with the help of socket programming. In socket programming the node which accepts request is server and the node that sends request is client.

**3. Design**

* 1. **Designing and Development**

In designing and development there are both the encoding and decoding of the input text file as well as devising a way to establish a connection between server and client using sockets.

Encoding - To encrypt the input text file such that it takes less space.

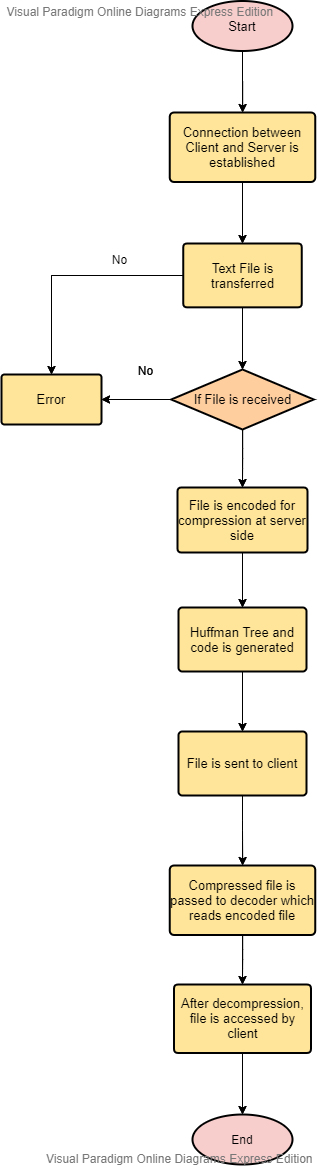
Decoding - To decrypt the encoded file back into the original file such that there is no loss of data.

Establishing Connection – One socket or a node listen to a particular port to establish a connection with other node using the IP address. Server is usually the listener socket while client reaches out to the server.

**3.1.1. Methodology**

* Firstly, connection between the client and server is established using socket programming. The server opens up a port number to which client will connect for transmission of text file.
* When the connection is established the sample text file is transferred from client to server.
* Server encrypts the received text file and then scans the entire file and stores the contents in the buffer.
* The encrypted file is then transmitted to the client which decrypts the content and is further transmitted to encoder.
* The frequency count of each character is calculated and store it in a table in ascending order.
* To compress the file, prepare the Huffman tree and generate the Huffman code for each character.
* Then the compressed file is passed to the decoder which reads the Huffman encoded file with one binary digit at a time to retrieve the original text.
* After the complete decompression, decompressed file (original contents of file) is generated which can be accessed by the client

**Flowchart**

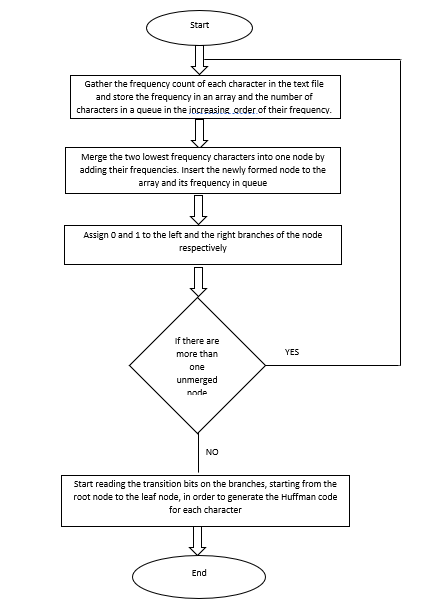


1. **Implementation**

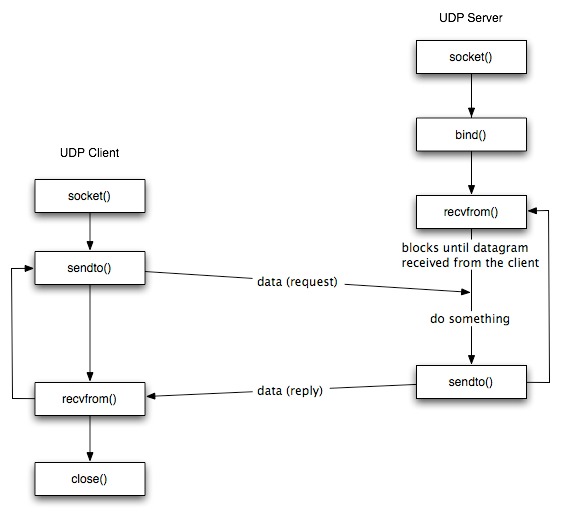
**4.1 Algorithm**

**FLOW CHART**

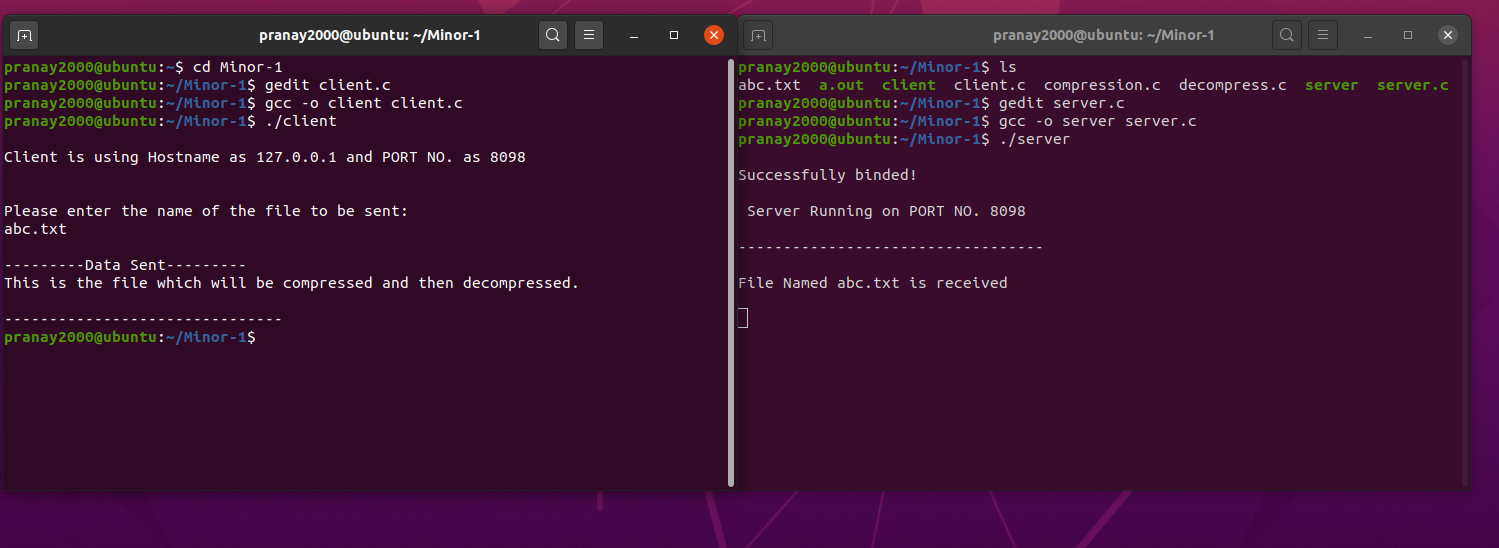
**Huffman Encoding**

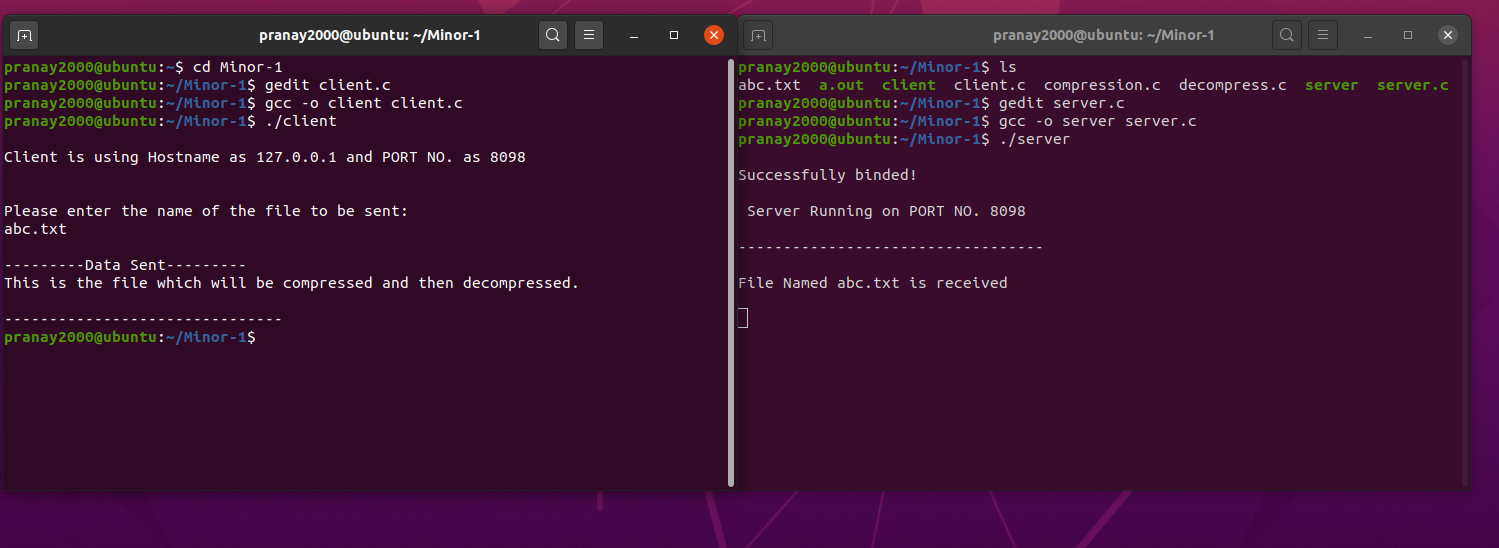


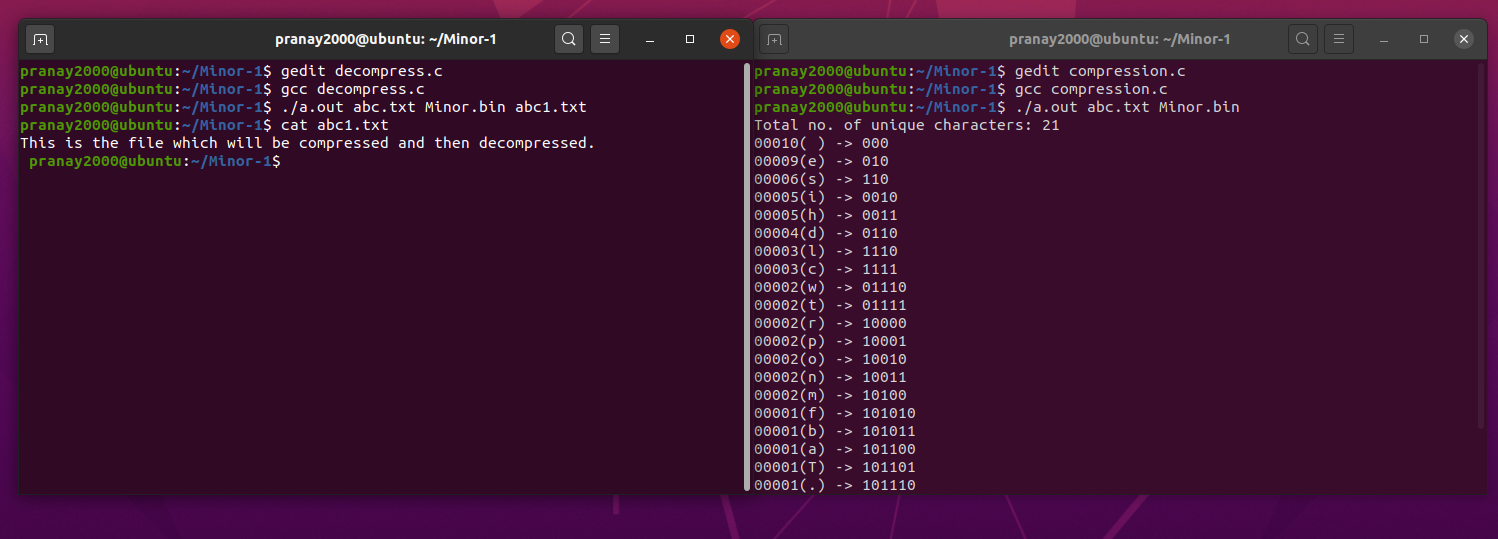
**Socket Programming**

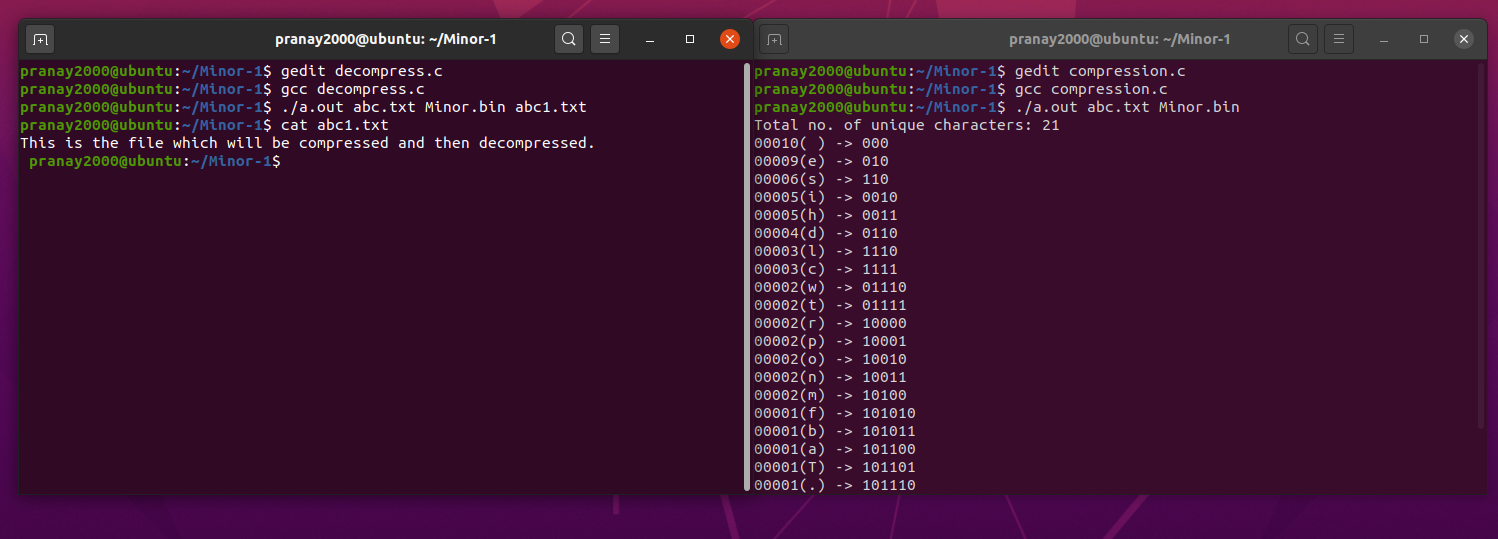


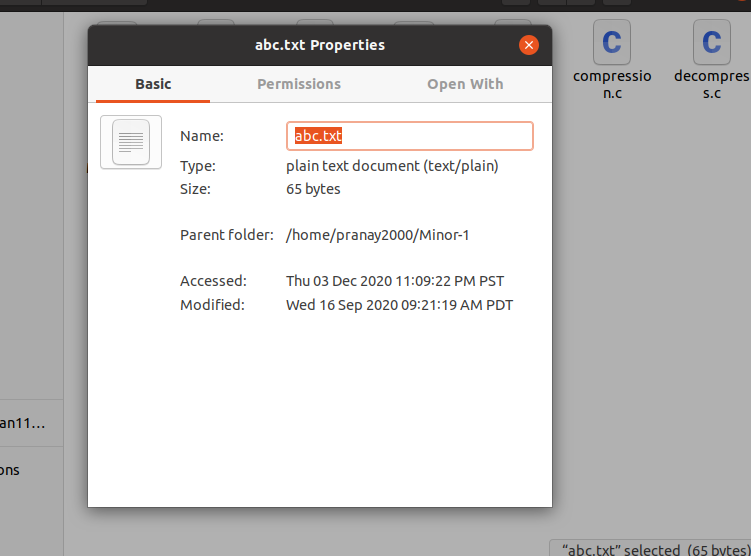
**5) Output Screens**

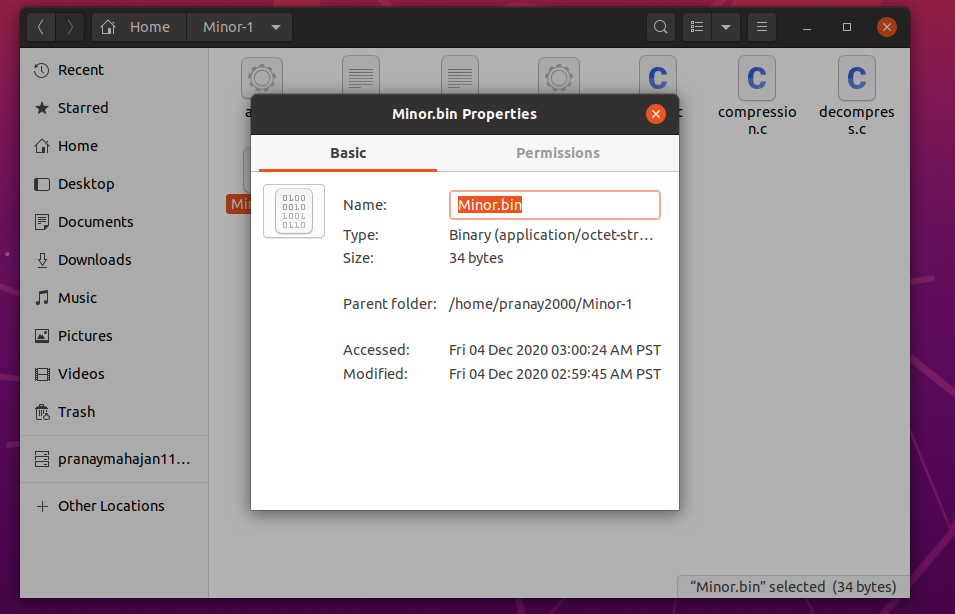












1. **Limitations and Future Enhancements**

**6.1 Limitations**

* This project deals only with text files.

**6.2 Enhancement**

Some of the possible amendments and improvements to be carried out in future are:

* Include transfer of other files such as jpg, mp4 etc.
* Include encryption algorithm for secure transmission of files.

1. **Conclusion**

The conclusion of the project is that huffman encoding and decoding had been successfully implemented. Apart from this, our other objective i.e. to transfer a file using socket programming has also been achieved.

**References**

**[1]** [**https://www.edureka.co/blog/socket-programming-in-java/**](https://www.edureka.co/blog/socket-programming-in-java/)

**[2]** [**https://www.kodefork.com/learn/algorithms/greedy-algorithms/**](https://www.kodefork.com/learn/algorithms/greedy-algorithms/)

**[3]** [**https://www.techiedelight.com/huffman-coding/**](https://www.techiedelight.com/huffman-coding/)

**[4]** [**https://medium.com/stantmob/data-compression-with-huffman-coding-ad7bcb07c5d5**](https://medium.com/stantmob/data-compression-with-huffman-coding-ad7bcb07c5d5)**.**

**[5]** [**https://books.google.co.in/books?id=Thealgorithmdesign**](https://books.google.co.in/books?id=Thealgorithmdesign)