INTERNATIONAL JOURNAL OF
RECENTT RENDS IN MULTI DISCIPLINARY RESEARCH
https://www.doi.org/10.59256/ijrtmr.20240401009
January-February 2024, Vol 4(01), 54-57.
ISSN No: 2583-0368

File Packer-Un packer System

Sampada Kulkarni[1], Tejas Ghadge[2], Ashutosh Mishra[3], Sameer Dhane[4], Pranav Jadhawar[5]

[1]Assistant Professor, Dept. of I.T., P.E.S. Modern College of Engineering, Savitribai Phule Pune University, Maharashtra, India.

[2,3,4,5]Dept. of I.T., P.E.S. Modern College of Engineering, Savitribai Phule Pune University, Maharashtra, India.

OPEN ACCESS

Article Citation:
Sampada Kulkarni[1], Tejas Ghadge[2], Ashutosh Mishra[3], Sameer Dhane[4], Pranav Jadhawar[5], "File Packer-Unpacker System", International Journal of Recent Trends In Multi disciplinary Research, January-February 2024, Vol 4(01), 54-57.

Published by 5th Dimension Research Publication

Abstract: The File Packer/ Un packer Application is a software designed to streamline the process of compressing and decompressing files and folders. In the digital age, the need for efficient data storage and transfer is paramount. This Application aims to address this need by providing a versatile and user-friendly tool for packing multiple files and directories into a single compressed archive, as well as extracting them when required.

The application features a robust user interface that allows users to select files and folders for packing and specify compression settings. It supports various compression algorithms, such as ZIP, GZIP, and RAR, offering flexibility to accommodate different user preferences and requirements. Additionally, the application provides options for password protection and encryption to enhance data security [1].

In a world where data management and security are of paramount importance, Our File Packer/Unpacker Application aims to simplify and enhance the process of file compression and decompression. This Application offers a versatile and secure solution for individuals and organizations seeking to optimize their data storage and transfer operations.

Key Word: File Management, Lempel-Ziv-Welch (LZW), Data security, Data storage, Achive Files

1. Introduction

The File Packer Unpacker Application is a comprehensive software Endeavor dedicated to optimizing file management through efficient compression and decompression processes. This initiative addresses the growing need for streamlined data storage and transfer solutions. By amalgamating multiple files into a singular archive, the file packer component significantly reduces overall data size, contributing to improved storage efficiency and facilitating smoother file transportation. The unpacker counterpart ensures the effortless restoration of the original content, maintaining the integrity of the compressed files.

A key focus of our project lies in user accessibility, incorporating intuitive interfaces that simplify the packing and unpacking operations. The underlying algorithms are designed for optimal performance, ensuring quick and reliable processing. This Application is particularly valuable in scenarios whereconsiderations such as limited storage capacity, bandwidth constraints, or expedited data transfer are paramount.

Overall, the File Packer Unpacker application addresses contemporary challenges in file management, offering a versatile and user-centric solution for individuals and organizations seeking to enhance the efficiency of their data handling processes.

2. Related Work

The File Packer Unpacker Application builds upon a foundation of related work in the field of file compression and decompression tools. Preceding projects, such as WinZip and WinRAR, have laid the groundwork for efficient file packaging and extraction. These tools have long been instrumental in reducing file sizes for storage and transmission, providing inspiration for the development of the File Packer Unpacker.

Furthermore, advancements in open-source projects like 7-Zip have influenced the design principles of the File Packer Un packer, emphasizing the importance of robust algorithms and compatibility across various file formats. Learning from the strengths and weaknesses of existing solutions, our application aims to offer an improved user experience with a focus on intuitive interfaces and optimized performance.

Additionally, research on compression algorithms, such as Lempel-Ziv-Welch (LZW) [2] and deflate, has contributed to the theoretical underpinnings of the File Packer Un packer's compression mechanisms. Studying these algorithms has allowed the project to implement strategies that balance compression ratios with processing speed. In summary, the File Packer Unpacker project is intricately connected to the evolution of file compression tools, drawing inspiration from established solutions while introducing innovations in user interface design, performance optimization, and compression algorithms.

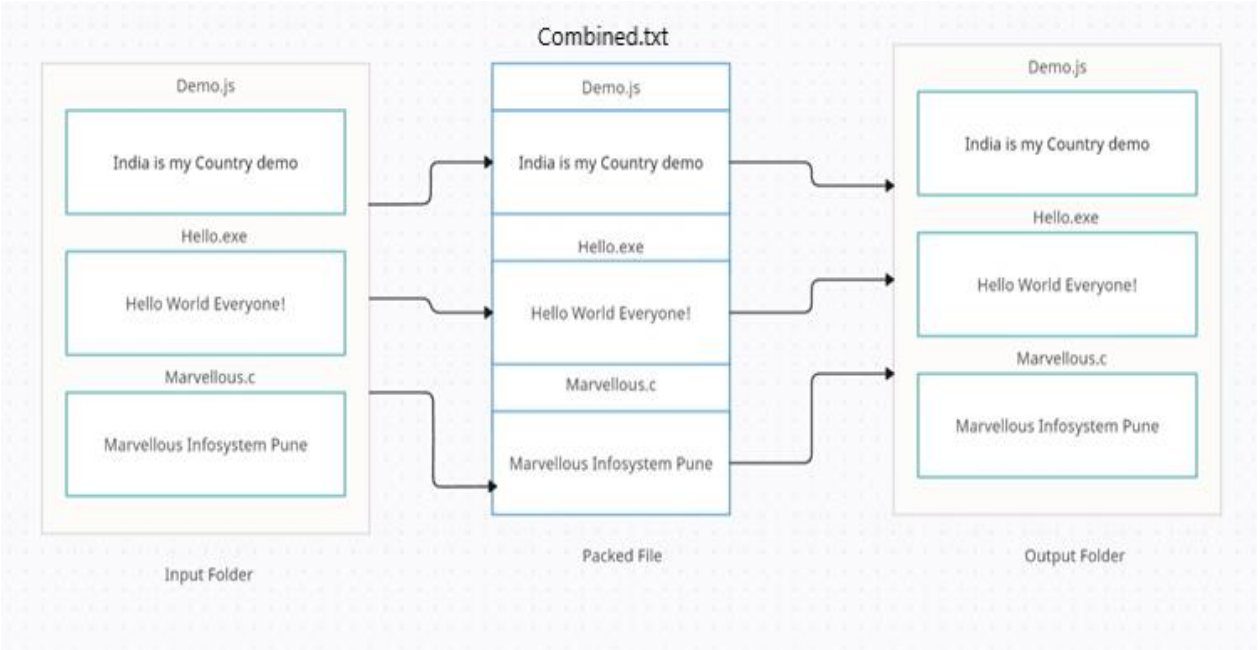
3. Proposed Architecture

The File Packer Unpacker architecture is designed as a modular and extensible system, consisting of two main components: the Packer and the Unpacker modules. The Packer module incorporates a hybrid compression algorithm, to efficiently reduce file sizes. It employs dynamic dictionary management and parallel processing for optimal performance. The Unpacker module seamlessly reverses the process, reconstructing the original files from the compressed data.

Both modules share a common core responsible for file I/O operations [3], ensuring compatibility and consistency across the packing and unpacking processes. The architecture emphasizes a user-friendly interface for ease of interaction.

Additionally, the project allows for future expansion by incorporating support for various file formats and potential integration of advanced compression techniques. The modular design facilitates extensibility, enabling developers to enhance or replace specific components as needed. Overall, the File Packer Unpacker architecture prioritizes efficiency, adaptability, and user accessibility in the realm of file compression and decompression.

a) **Product Features**



Login Screen	User enter his details and enters
Input Folder	Contains Multiple files.
Packed File	Single File contains multiple files.
Encryption	Single File contain data with encryption.

b) User Roles and Rights

1. User Friendly interface.
2. Cross Platform compatibility
3. Security Measures.
4. Verifies and confirms Passwords.
5. Access the data

4. Proposed System

The operation of the project is initially to be designed for windows operating systems. But, because we are using java for this application development Hence, It is platform Independent.

Os: Any Operating System is Valid.

Languages used: Java (1.8 and above)

Software used: Apache Net Beans, and modelling software

Hardware used: PC/Laptop/ tablet with 4 gigs ram, functional keyboard /mouse/touch screen

Disk space: Disk space of 500MB and another 1GB free space required.

5. Proposed Algorithm

TheFilePackerUnpackerprojectproposesahybridcompressionalgorithmthat combine elements of both Lempel-Ziv-Welch (LZW) and Huffman coding [2]. This algorithm aims to achieve an optimal balance between compression efficiency and processing speed.

During the packing phase, the LZW component is employed to identify and replace repeated sequences of data with variable-length codes, effectively reducing redundancy in the file. Simultaneously, Huffman coding is applied to further compress the data by assigning shorter codes to more frequent symbols, enhancing the overall compression ratio.

In the unpacking phase, the algorithm intelligently utilizes the stored LZW and Huffman codes to reconstruct the original file. This process involves decoding the variable-length codes and efficiently reconstructing the compressed data.

To enhance apt ability, the algorithm incorporates dynamic dictionary management; allow ingot to update and optimize the dictionary during both packing and unpacking basedon the encounter reddat apatterns. This adaptability ensures improved compression performance across various types of files.

Additionally, the project considers parallel processing techniques to enhance the overall speed of the compression and decompression processes, making efficient use of modern multi- core processors.

By combining the strengths of LZW and Huffman coding with dynamic dictionary management and parallel processing, the proposed algorithmaims to delivera versatile and high- performance solution for the File Packer Unpacker project.

6. Procedure of Packing and Unpacking

Packing Activity:

- In case of Packing activity, we accept directory name and file name from user [4].
- We have to create new regular file as the name specified by the user.
- Now open the directory and traverse each file from that directory. In newly created file write Metadata as header and actual file data in sequence.
- While writing data perform encryption.
- Each name of file, its size and checksum should be written in log file which gets create din system directory.
- After packing display packing report.

Unpacking Activity:

- In case of unpacking activity, we accept packed file name from user. For authentication of packed file use any logic like Magic Number.
- Open the packed file in read mode and perform below activity as Read header.
- From the name specified in header create new file.
- Write data into newly created file from packed file.
- Repeat all above step still we reached at end of the file unpacked file.
- After unpacking display unpacking report.

This step provides a high-level overview of the essential steps involved in both packing and unpacking processes for the file packer Un packer project. Actual implementation details may vary based on specific requirements and the chosen programming language.

7. Conclusion

In conclusion, the File Packer Unpacker project introduces a versatile and efficient solution for file compression and decompression, leveraging a hybrid algorithm combining Lempel-Ziv-Welch (LZW) and Huffman coding. This project addresses the contemporary demand for streamlined data storage and transfer, emphasizing user-friendly interfaces and optimized performance.

The hybrid compression algorithm demonstrates promising results in reducing file sizes while maintaining a balance between compression ratios and processing speed. By incorporating dynamic dictionary management and parallel processing techniques, the algorithm adapts to diverse file types and harnesses the capabilities of modern hardware architectures.

Additionally, considering the integration of error-checking mechanisms enhances the robustness of the file unpacking process, ensuring data integrity during decompression.

Furthermore, the application benefits from incorporating support for a wider range of file formats and introducing

cross-platform compatibility. Extending the capabilities to handle multimedia files, archives, and various data structures would broaden the utility of the File Packer Un packer, making it more versatile for different user scenarios.

Collaboration with the open-source community and continuous refinement based on user feedback would be integral to the project's evolution. Conducting rigorous testing and benchmarking against existing compression tools would validate the algorithm's performance and identify areas for further optimization.

In summary, the File Packer Un packer project stands as a robust foundation for efficient file management, and its future evolution holds the promise of even greater versatility, performance, and adaptability in the dynamic landscape of data compression and decompression.

References

1. *Irfan Shaikh, Pritesh Bafna, Prof. Mr. S. R Lahane, "File Sharing System," International Journal of Scientific and Research Publication, Volume 3, Issue 6, ISSN 2250-3153, p. 4, June 2022.*

2. *K. Muthuchamy, P. Ravi, Dr. A. Ashok Kumar, "A Study On Various Data Compression Types And Techniques," International Journal of Computer Science & Communication, vol. Volume 6, no. Issue 2, p. 8, April-September 2015.*

3. *Atiya Mumtaz, "Survey Paper on Adding System Call in Linux 3.2+ & 3.16," International Journal of Science and Research (IJSR), vol. Volume 6, no. Issue 1, p. 3, January 2017.*

4. *Nikhil Kumar, Akshay Bansal, Kartik Singhal, Pratham Sharma, Dr. Vinesh Kumar, "File Management System," International Journal of Computer Science and Information Technology Research, vol. 8, no. 2, p. 5, April - June 2020.*