RV College of Engineering

Experiential Learning - Phase 1

DISK SPACE ANALYZER

Presented by :
Vidwath H Hosur– 1RV22CS231
Shriniwas Maheshwari – 1RV22CS194

Go, change the world

RV College of Engineering

Introduction

A Disk Space Analyzer is a utility tool that helps users understand how disk space is being utilized on their computer. It scans the hard drive (or a specific folder) and provides a detailed view of which folders and files are using space1. This allows users to make informed decisions about what to remove to quickly free up space. It aims to provide a comprehensive solution for analyzing and managing disk space usage on computer systems.

Problem Statement

Create a tool to scan and display disk space usage, helping users identify large files and directories.

Implementation :

1. **Directory Traversal and File System Monitoring**: Implement functions to traverse the directory structure recursively. Use system calls or platform-specific APIs to monitor file system events such as file creations, modifications, and deletions.

2. **Data Collection and Analysis**: Collect information about directories, subdirectories, and files, including their sizes and attributes. Aggregate disk space usage statistics for each directory and file.

3. **User Interface**: Develop a user-friendly interface to interact with the Disk Space Analyzer. Include options for users to start the analysis, view results, and configure settings. Design intuitive displays to present disk space usage information, such as directory hierarchies, file sizes, and trends.

4. **Identification of Large Files and Directories:** Define thresholds to identify large files and directories based on size. Implement algorithms to flag files and directories that exceed specified thresholds.

5. **Reporting and Visualization:** Generate detailed reports and summaries of disk space usage metrics. Visualize disk space usage data using charts, graphs, and tables. Provide sorting options to allow users to sort results based on size, name, or other criteria.

6. **Configuration and Customization:** Allow users to configure settings such as scan depth, excluded directories, and backup schedules. Provide filtering options to focus on specific directories or file types.

API TOOLS USED:

```
// Function to traverse directories and collect disk space usage
information
void traverseDirectory(char* path);

// Function to monitor file system events and update disk space usage
information
void monitorFileSystem();

// Function to analyze disk space usage and identify large files and
directories
void analyzeDiskSpace();

// Function to generate disk space usage reports and visualize data
void generateReports();

// Function to handle user interactions and display the user interface
void displayUI();
```

Applications in outputs :

- Implement features like graphical visualization of disk usage using external libraries.
- Add options to exclude certain file types or directories from the analysis.
- Include a feature to save the results to a file for later reference.

RELEVANCE

1. File System APIs :

- File System Traversal: Utilize file system APIs provided by the operating system to traverse directories and access file attributes.
- File System Monitoring: Utilize APIs for file system monitoring to detect changes in directories and files, allowing the analyzer to react to modifications in real-time.

2. Process Management :

- Multithreading: Employ multithreading to perform disk space analysis concurrently with other operations, enhancing responsiveness and efficiency.
- Priority Management: Utilize process priorities to allocate resources effectively, ensuring that disk space analysis tasks are performed with appropriate precedence.

3. Memory Management :

- Buffering: Implement buffering techniques to optimize disk I/O operations, enhancing the performance of file reading and traversal.
- Caching: Utilize operating system-level caching mechanisms to store frequently accessed file.

4. Inter-Process Communication (IPC):

- Event Notification: Implement IPC mechanisms such as signals or message queues to facilitate communication between the disk space analyzer and other system components.
- Data Sharing: Enable sharing of data between different processes involved in disk space analysis, enhancing collaboration and efficiency.

5. File System Security:

- Access Control: Leverage file system security mechanisms provided by the operating system to enforce access control policies, ensuring that the disk space analyzer operates within the confines of user permissions.
- Encryption: Utilize file encryption mechanisms provided by the operating system to protect sensitive data stored or accessed by the disk space analyzer.

6. Error Handling and Recovery:

- Error Reporting: Utilize system-level error reporting mechanisms to provide informative and actionable error messages to users, aiding in troubleshooting and recovery efforts.
- Fault Tolerance: Implement fault tolerance mechanisms to ensure graceful recovery from errors or failures encountered during disk space analysis, minimizing data loss or disruption.

DSA COMPONENT

1. Directory Structure Representation:

- Tree Data Structure: Represent the directory structure using a tree, where each node represents a directory, and its children represent subdirectories and files.
- Node Attributes: Include attributes such as directory name, size, and references to child nodes.

2. File Information Management:

- Linked Lists or Arrays: Maintain lists or arrays to store file information within directories.
- File Attributes: Store attributes such as file name, size, type, and modification date.

3. Memory Management:

- Dynamic Data Structures: Use dynamic data structures such as linked lists or dynamic arrays to accommodate varying numbers of directories and files.
- Memory Allocation Algorithms: Employ memory allocation algorithms to optimize memory usage and avoid memory fragmentation.

4. File System Monitoring:

- Event Queues: Implement queues to store file system events such as file creations, modifications, and deletions.
- Priority Queues: Utilize priority queues to prioritize file system events based on urgency or importance.

5. Error Handling and Recovery:

- Exception Handling: Implement exception handling mechanisms to gracefully handle errors encountered during disk space analysis.
- Recovery Algorithms: Design algorithms to recover from errors or failures and resume analysis without data loss or corruption.

LITERATURE SURVEY

- **iHDViewer: A Visualization Tool for Tracking HD** This paper presents a new interactive visualization technique named iHard Disk Viewer (iHDViewer), which aims to display file details and their movement histories, as well as predict the user's activities through visualization results.
- **Disk Space Analyzer Reviews:** This G2 page provides overviews and user reviews of various disk space analyzers, helping users choose the right tool for their needs. (G2)
- **Projecting Disk Usage Based on Historical Trends in a Cloud Environment June 2012DOI:**10.1145/2287036.2287050Conference: Proceedings of the 3rd international workshop on Scientific cloud computing Authors: Murray Stokely, Amaan Mehrabian, Christoph Albrecht, Francois Labelle -"measured the disk space usage, I/O rate, and age of stored data for thousands of different engineering users and teams."

THANK YOU