|  |  |  |
| --- | --- | --- |
| Abstract | | |
|  | | |
| TwoLevelDirectoryProgram | | |
|  |  | Two Level Directory in OS - GeeksforGeeksPhoto of a toucan in a green forest |
|  | **TwoLevelDirectoryProgram** |

The two-level directory system organizes files in a hierarchical structure with a global root directory and separate subdirectories for each user.

This approach mitigates naming conflicts by isolating each user’s files in distinct directories and simplifies management by centralizing user directories under a common root.

The implementation illustrated here uses linked lists to simulate the two-level directory structure, providing functions to create files, display directory contents, and search for files within a user's directory.

This code demonstrates the essential concepts behind multi-user file systems in operating systems.

* #include <stdio.h>  
  *Includes functions for input/output (printf, scanf).*
* #include <stdlib.h>  
  *Includes functions for memory management and utility functions (malloc, exit).*
* #include <string.h>  
  *Provides functions for string operations (strcpy, strcmp).*
* #define MAX\_NAME 50  
  *Sets the maximum length for file and user names.*

**Data Structures:**

* **FileNode structure:**
  + char filename[MAX\_NAME];  
    *Stores the file name.*
  + FileNode\* next;  
    *Pointer to the next file node, enabling a linked list of files.*
* **UserDirectory structure:**
  + char username[MAX\_NAME];  
    *Stores the user name.*
  + FileNode\* fileList;  
    *Points to the head of a linked list containing the user's files.*
  + UserDirectory\* next;  
    *Pointer to the next user directory in the global list.*

**Global Variable:**

* UserDirectory\* root = NULL;  
  *Initializes the global root pointer for the user directory list.*

**Function Prototypes:**

* Declares functions:
  + createUserDirectory: Creates a new user directory.
  + addUserDirectory: Adds a user directory to the global list.
  + createFile: Adds a file to a specific user's directory.
  + listFiles: Lists all files for a user.
  + searchFile: Searches for a specific file in a user's directory.
  + findUser: Finds a user directory by username.

**Main Function:**

* int main() {
  + Declares variables (choice, username, filename).
  + Pre-creates user directories for "alice" and "bob" using createUserDirectory and addUserDirectory.
  + Enters an infinite loop displaying a menu.
  + Uses a switch-case to process user input:
    - **Case 1:** Prompts for a username and filename, then calls createFile.
    - **Case 2:** Prompts for a username, then calls listFiles.
    - **Case 3:** Prompts for a username and filename, then calls searchFile.
    - **Case 4:** Exits the program using exit(0).
  + Returns 0 to indicate successful termination.
  + Prints whether the file is found or not.

**Function Definitions:**

1. **createUserDirectory:**
   * Allocates memory for a new user directory.
   * Copies the username into the directory.
   * Initializes fileList and next pointers to NULL.
   * Returns the new directory pointer.
2. **addUserDirectory:**
   * Checks if the global root is NULL; if yes, assigns the new directory as root.
   * Otherwise, traverses the linked list and appends the new directory at the end.
3. **findUser:**
   * Traverses the global directory list.
   * Uses strcmp to compare each directory’s username with the provided one.
   * Returns the matching directory or NULL if not found.
4. **createFile:**
   * Calls findUser to locate the user directory.
   * If the user is found, allocates memory for a new FileNode.
   * Copies the filename and initializes its next pointer.
   * Appends the FileNode to the user's file list.
   * Prints a confirmation message.
5. **listFiles:**
   * Calls findUser to locate the user directory.
   * If found, iterates through the file linked list and prints each filename.
   * Informs the user if no files are found.
6. **searchFile:**
   * Calls findUser to locate the user directory.
   * Iterates through the file linked list comparing filenames.
   * Prints whether the file is found or not.