#include <iostream>

#include <fstream>

#include <string>

#include <vector>

class FileManager {

public:

void createUserFolder(const std::string& username) {

std::string folderPath = "files/" + username;

if (!folderExists(folderPath)) {

createFolder(folderPath);

std::cout << "Folder for user '" << username << "' created at: " << folderPath << std::endl;

} else {

std::cout << "Folder for user '" << username << "' already exists at: " << folderPath << std::endl;

}

}

void importFile(const std::string& username, const std::string& filename) {

std::string sourcePath = filename;

std::string destinationPath = "files/" + username + "/" + filename;

if (fileExists(sourcePath)) {

copyFile(sourcePath, destinationPath);

std::cout << "File '" << filename << "' imported to user '" << username << "' folder." << std::endl;

} else {

std::cout << "Error: File '" << filename << "' not found." << std::endl;

}

}

void openFile(const std::string& username, const std::string& filename) {

std::string filePath = "files/" + username + "/" + filename;

if (fileExists(filePath)) {

std::ifstream file(filePath);

if (file.is\_open()) {

std::string line;

while (getline(file, line)) {

std::cout << line << std::endl;

}

file.close();

} else {

std::cout << "Error: Unable to open file '" << filename << "'." << std::endl;

}

} else {

std::cout << "Error: File '" << filename << "' not found." << std::endl;

}

}

private:

bool folderExists(const std::string& path) {

struct stat info;

return stat(path.c\_str(), &info) == 0 && info.st\_mode & S\_IFDIR;

}

bool fileExists(const std::string& path) {

std::ifstream file(path);

return file.good();

}

void createFolder(const std::string& path) {

#ifdef \_WIN32

\_mkdir(path.c\_str());

#else

mkdir(path.c\_str(), S\_IRWXU | S\_IRWXG | S\_IROTH | S\_IXOTH);

#endif

}

void copyFile(const std::string& source, const std::string& destination) {

std::ifstream sourceFile(source, std::ios::binary);

std::ofstream destinationFile(destination, std::ios::binary);

destinationFile << sourceFile.rdbuf();

sourceFile.close();

destinationFile.close();

}

};

int main() {

FileManager fileManager;

std::string adminUsername = "admin";

std::string userUsername = "user";

fileManager.createUserFolder(adminUsername);

fileManager.createUserFolder(userUsername);

fileManager.importFile(adminUsername, "file1.txt");

fileManager.importFile(userUsername, "file2.txt");

fileManager.openFile(adminUsername, "file1.txt");

fileManager.openFile(userUsername, "file2.txt");

return 0;

}