

```

Консоль отладки Microsoft Visual Studio
Path length: 32
Path content: [D:\Univer\cpp\Lab4\TestTask1.txt]
File exists. Fetching attributes...
=== File Attributes: D:\Univer\cpp\Lab4\TestTask1.txt ===
File Size: 29 bytes
Creation Time: 29/11/2024 3:17:2
Last Access Time: 29/11/2024 10:50:2
Last Modification Time: 29/11/2024 8:34:59
Checking large file for reading: D:\Univer\cpp\Lab4\LargeTestFile.dat
Large file exists. Starting benchmarks...
fopen/fread: Read 104857600 bytes in 0.553 seconds.
Windows API: Read 104857600 bytes in 0.0457044 seconds.
Processing files asynchronously...
Async file processing is not yet implemented.

D:\Univer\cpp\Lab4\x64\Debug\Lab4.exe (процесс 16856) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

Було використано функції для зчитування атрибутів файлу, таких як дата створення, останнє використання, розмір тощо. А з консолі виведено наступну інформацію:

Виведена інформація:

Атрибути файлу TestTask1.txt: довжина шляху, дата створення, останньої модифікації тощо.

Продуктивність читання файлу LargeTestFile.dat:

fopen/fread: 0.553 секунд.

Windows API: 0.0457044 секунд.

Асинхронна обробка файлів поки не реалізована.

Для того, щоб створити потрібний файл розміром 100мб(згідно завдання) було окремо ще реалізований код для штучного створення файлу:

```

bigfile (Глобальная область)
#include <iostream>
#include <fstream>
#include <memory>
#include <string>

void createLargeTestFile(const std::string& filePath, size_t fileSizeMB) {
    std::ofstream outFile(filePath, std::ios::binary);
    if (!outFile) {
        std::cerr << "Failed to create test file: " << filePath << std::endl;
        return;
    }

    const size_t bufferSize = 1024 * 1024; // 1 MB
    std::unique_ptr<char[]> buffer(new char[bufferSize]);
    memset(buffer.get(), 'A', bufferSize);

    for (size_t i = 0; i < fileSizeMB; ++i) {
        outFile.write(buffer.get(), bufferSize);
    }

    outFile.close();
    std::cout << "File created successfully: " << filePath << " (" << fileSizeMB << " MB)" << std::endl;
}

int main() {
    std::string filePath = "C:\\Users\\vico\\OneDrive\\Рабочий стол\\Учеба\\C++\\bigfile\\LargeTestFile.dat";
    size_t fileSizeMB = 100;

    // файл
    createLargeTestFile(filePath, fileSizeMB);

    return 0;
}

```

Підсумовуючи, використання Windows API значно прискорює процес читання великих файлів порівняно зі стандартними функціями C++ (fopen/fread). Час читання через Windows API (0.0457 секунд) є суттєво меншим за час читання через fopen/fread (0.553 секунд). Програма продемонструвала базову підготовку до асинхронної обробки файлів, але ця функція не була реалізована