

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення



ЗВІТ

до лабораторної роботи № 8
на тему: «Управління файловою системою.»
з дисципліни: «Операційні системи»

Лектор:

ст. викладач кафедри ПЗ
Грицай О. Д.

Виконав:

ст. гр. ПЗ-22
Чаус О. М.

Прийняла:

ст. викладач кафедри ПЗ
Грицай О. Д.

« ____ » _____ 2022 р.

Σ= _____

Тема роботи: Управління файловою системою.

Мета роботи: Ознайомитися з файловими системами операційних систем Windows та LINUX.

Теоретичні відомості

Файлова система – це система правил, згідно яких операційна система забезпечує зберігання даних на диску. Ці правила визначають формат зберігання даних, їх розміщення на диску, адресацію і ідентифікацію. Кожен комп'ютерний файл зберігається на цифровому носії із певною місткістю. Накопичувач можна уявити як лінійний простір для читання і запису цифрової інформації. Кожен байт даних на ньому має своє зміщення від початку сховища, відомий як адреса, яка використовується для посилання на цей файл. Накопичувачі для посилання на байт інформації можуть використовувати адресацію у вигляді пари - сектор і зміщення. Загалом, файлова система це структуроване представлення даних та набір метаданих, що описують ці дані. Вона записується до сховища під час операції форматування. Зазвичай ця структура оперує блоками, а не секторами. Блоки ФС – це групи секторів, за допомогою яких оптимізується адресація сховища. Файли, як правило, зберігаються на початку блоку та займають цілі блоки. Крім даних користувача, файлова система також містить власні параметри (наприклад, розмір блоку), файлові дескриптори (такі як його розмір, розташування, фрагменти і т.д.), імена та ієрархію каталогів. Вона також може зберігати інформацію про безпеку, розширені атрибути та інші параметри. Щоб відповідати різноманітним вимогам користувачів, таким як продуктивність сховища, стабільність і надійність, розробляється безліч типів (або форматів) файлових систем, здатних більш ефективно виконувати певні функції.

Завдання для виконання лабораторної роботи

1. Створити за допомогою API функцій файл для запису результатів виконання лабораторної роботи №3 (відповідно до свого варіанту).
2. Створити файл для запису результатів виконання лабораторної роботи №4 (відповідно до свого варіанту).
3. Реалізувати зміну прав доступу користувача до файлу.
4. Результати виконання роботи відобразити у звіті.

Хід роботи

1. Реалізував запис у файл для лабораторної роботи №3.

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <string>
#include <Windows.h>

struct Arctan {
    double x;
    double arctanx;
};

std::string calculate(double A, double B, int steps, double eps) {
    std::cout << std::setprecision(10);
    std::string res = "";
    std::vector<Arctan> info(steps + 1);
    double step = (B - A) / steps;
    for (std::size_t i = 0; i < info.size(); i++) {
        info[i].x = A + (step * i);
        double sum = 0;
        double iter_value = eps + 1;
        int k = 0;
        while (fabs(iter_value) > eps) {
```

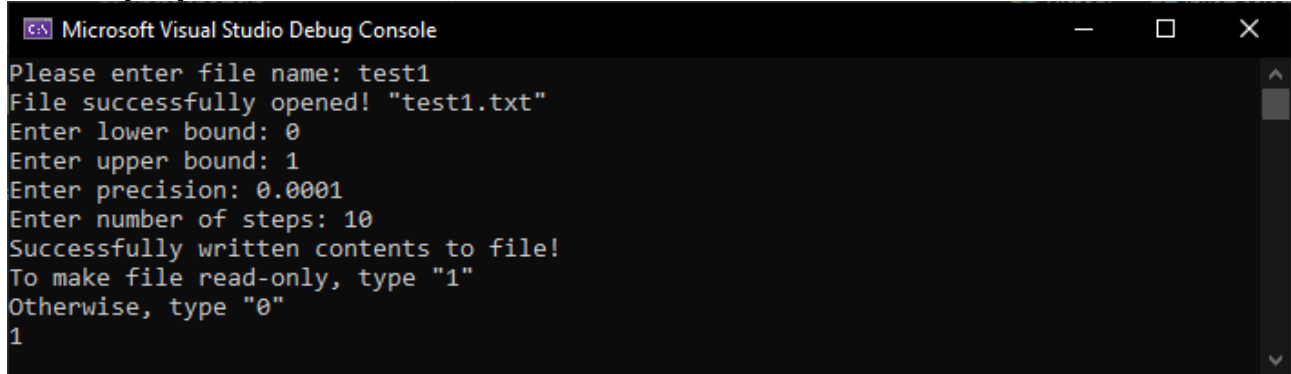
```

        iter_value = (pow(-1, k) * pow(info[i].x, 1 + 2 * k)) / (1 + 2 * k);
        k++;
        sum += iter_value;
        if (k > 10) break;
    }
    info[i].arctanx = sum;
    res.append(std::to_string(info[i].x) + "\t" + std::to_string(info[i].arctanx)
+ "\n");
    }
    return res;
}

int main() {
    std::string filename;
    double a, b, eps;
    unsigned steps;
    std::cout << "Please enter file name: ";
    std::cin >> filename;
    filename.append(".txt");
    HANDLE file = CreateFileA(filename.c_str(), GENERIC_WRITE, NULL, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, NULL);
    if (file != INVALID_HANDLE_VALUE) {
        std::cout << "File successfully opened! \"" << filename << "\"\n";
        while (true) {
            std::cout << "Enter lower bound: ";
            std::cin >> a;
            std::cout << "Enter upper bound: ";
            std::cin >> b;
            if (a < b)
                break;
            else std::cout << "Lower bound has to have smaller value than upper
bound!\n";
        }
        std::cout << "Enter precision: ";
        std::cin >> eps;
        std::cout << "Enter number of steps: ";
        std::cin >> steps;
        std::string to_write = calculate(a, b, steps, eps);
        if (WriteFile(file, to_write.c_str(), to_write.size(), NULL, FALSE))
            std::cout << "Successfully written contents to file!\n";
        system(("type" + filename).c_str());
        std::cout << "To make file read-only, type \"1\"\n0otherwise, type \"0\"\n";
        int ans;
        std::cin >> ans;
        if (ans == 1)
            SetFileAttributesA(filename.c_str(), FILE_ATTRIBUTE_READONLY);
        CloseHandle(file);
    }
    else std::cout << "Failed to create file! ;(\n";
    return 0;
}

```

Вивід програми:



```
Microsoft Visual Studio Debug Console
Please enter file name: test1
File successfully opened! "test1.txt"
Enter lower bound: 0
Enter upper bound: 1
Enter precision: 0.0001
Enter number of steps: 10
Successfully written contents to file!
To make file read-only, type "1"
Otherwise, type "0"
1
```

2. Реалізував запис у файл для лабораторної роботи №4

```
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>
#include <map>
#include <cmath>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
struct Arctan {
    double x;
    double arctanx;
};

std::string calculate(double A, double B, int steps, double eps) {
    std::string res = "";
    std::vector<Arctan> info(steps + 1);
    double step = (B - A) / steps;
    for(std::size_t i = 0; i < info.size(); i++) {
        info[i].x = A + (step * i);
        double sum = 0;
        double iter_value = eps + 1;
        int k = 0;
        while(fabs(iter_value) > eps) {
            iter_value = (pow(-1, k) * pow(info[i].x, 1 + 2 * k)) / (1 + 2 * k);
            k++;
            sum += iter_value;
            if(k > 10) break;
        }
        info[i].arctanx = sum;
        res.append(std::to_string(info[i].x) + "\t" + std::to_string(info[i].arctanx) + "\n");
    }
    return res;
}

int main() {
    std::string filename;
    double a, b, eps;
    unsigned steps;
    std::cout << "Enter file name: ";
    std::cin >> filename;
    filename.append(".txt");
    int fd = open(filename.c_str(), O_WRONLY | O_CREAT | O_TRUNC, S_IRWXU);
    if(fd != -1) {
        std::cout << "File opened successfully! \"\" << filename << "\"\n";
        while (true) {
            std::cout << "Enter lower bound: ";
            std::cin >> a;
            std::cout << "Enter upper bound: ";
            std::cin >> b;
            if(a < b)
```

```

        break;
        std::cout << "Lower bound has to have lower value than upper bound!\n";
    }
    std::cout << "Enter precision: ";
    std::cin >> eps;
    std::cout << "Enter number of steps: ";
    std::cin >> steps;
    std::string to_write = calculate(a, b, steps, eps);
    if(write(fd, static_cast<const void*>(to_write.c_str()), to_write.size()) != -1) {
        std::cout << "Written to file successfully! Here are the contents of the file:\n";
        system(("cat " + filename).c_str());
    }
    std::map<std::string, int> str_mode {
        {"S_IRWXU", S_IRWXU},
        {"S_IRUSR", S_IRUSR},
        {"S_IWUSR", S_IWUSR},
        {"S_IXUSR", S_IXUSR},
    };
    int size;
    std::cout << "How many flags do you want to set? (max value is 3)\n";
    if(size != 0) {
        std::cin >> size;
        if(size > 3)
            size = 3;
        std::vector<int> modes{size};
        for(int i = 0; i < size; i++)
            std::string mode;
            while(true) {
                std::cout << "Enter desired mode flag.\n";
                std::cin >> mode;
                if(str_mode.find(mode) != str_mode.end()){
                    modes[i] = str_mode[mode];
                    std::cout << str_mode[mode] << "\n";
                    break;
                }
                std::cout << "Please enter correct flag.\n";
            }
        int final_mode = 0;
        for(int i = 0; i < size; i++)
            final_mode = final_mode | modes[i];
        std::cout << final_mode << "\n";
        if(fchmod(fd, final_mode) != -1)
            std::cout << "Changed mode successfully!\n";
    }
    return 0;
}
}

```

Вивід програми:

```
oleh@oleh-VirtualBox:~/os_lab8$ ./main
Enter file name: testss
File opened successfully! "testss.txt"
Enter lower bound: 0
Enter upper bound: 1
Enter precision: 0.0001
Enter number of steps: 10
Written to file successfully! Here are the contents of the file:
0.000000      0.000000
0.100000      0.099669
0.200000      0.197397
0.300000      0.291455
0.400000      0.380510
0.500000      0.463640
0.600000      0.540412
0.700000      0.610708
0.800000      0.674903
0.900000      0.735026
1.000000      0.808079
How many flags do you want to set? (max value is 3)
2
Enter desired mode flag.
S_IRUSR
256
Enter desired mode flag.
S_IWUSR
128
384
Changed mode successfully!
oleh@oleh-VirtualBox:~/os_lab8$ ls -l testss.txt
-rw----- 1 oleh oleh 198 Nov 17 11:17 testss.txt
```

Висновок: під час виконання лабораторної роботи ознайомився з файловими системами операційних систем Windows та LINUX.