Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Навчально-науковий комплекс «Інститут прикладного системного аналізу»

Кафедра системного проектування

Лабораторна робота № 3

з курсу «Операційні системи»

Виконала студентка

Групи ДА-02

Лесечко О. Р

Київ 2022

**Завдання:**

Написати утиліту, що починає роботу в заданому каталозі і спускається по дереву каталогів, записуючи по шляху розміри всіх файлів, що зустрілися їй. Закінчивши сканування каталогу, програма повинна роздрукувати гістограму розмірів файлів, використовуючи крок гістограми як параметр (наприклад, при кроці 1024, файли розміром від 0 до 1023 байт потрапляють в один інтервал, від 1024 до 2047 байт - в наступний інтервал і т.д. ).

**Код програми:**

#include <iostream>

#include <filesystem>

#include <map>

// класс Гистограмма, реализация построения гистограммы

class Histogram {

public:

/\* конструктор класса с аргуметом шага,

шаг - интервал в байтах \*/

Histogram(int step) {

this->step = step;

}

/\* сканирование каталога, метод начинает работу в заданном каталоге

и спускается по дереву каталогов, записывая по пути размеры

всех встретившихся ей файлов \*/

void traverseAllDirectoryFolders() {

try {

// рекурсивный обход каталога

for (const std::filesystem::path& path\_tmp : std::filesystem::recursive\_directory\_iterator(std::filesystem::current\_path())) {

// проверка на то файл это или нет

if (!std::filesystem::is\_directory(path\_tmp)) {

int sizeFile = std::filesystem::file\_size(path\_tmp); // размер файла

map[findKey(sizeFile)]++; // увеличить поле гистограммы на 1

}

}

}

catch (std::filesystem::filesystem\_error& e) {

std::cerr << e.what() << std::endl;

}

// для красивого построения гистограммы

auto it = map.end(); --it;

maxLength = length(it->first \* step) + length((it->first + 1) \* step);

}

/\* распечатать гистограмму размеров файлов,

используя шаг гистограммы в качестве парамет \*/

void printHistogram() {

// обход всех записей

for (auto it = map.begin(); it != map.end(); ++it) {

// все записи в этом блоке построения гистограммы

std::cout << "[" << it->first \* step << " - " << (it->first + 1) \* step << "] ";

int n = maxLength - (length(it->first \* step) + length((it->first + 1) \* step));

while (n > 0) {

n--;

std::cout << " ";

}

std::cout << " │";

for (int i = 0; i < it->second; i++) {

std::cout << "▀";

}

std::cout << "\n";

}

}

private:

/\* это ассоциативный контейнер, в котором хранятся элементы,

образованные комбинацией значения ключа(интервал в байтах)

и сопоставленного значения(количество встречающихся элементов в заданном интервале)

в определенном порядке \*/

std::map<int, int> map;

// шаг - интервал в байтах

int step = 1024;

// для красивого построения гистограммы

int maxLength = 1;

// определить в какой интервал входит размер файла

int findKey(int sizeFile) {

return sizeFile / step;

}

// для красивого построения гистограммы

int length(int x) {

if (x == 0) return 1;

return (log10(x) + 1);

}

};

int main() {

Histogram histogram(1024);

histogram.traverseAllDirectoryFolders();

histogram.printHistogram();

return 0;

}

**Результат роботи:**

Наприклад, я хочу побудувати гістограму для каталогу з моєю курсовою роботою щоб подивитися якого розміру файли там знаходяться.

Для цього мені треба зайти в потрібний мені каталог і за допомогою консолі запустити мою програму, приклад на малюнку нижче.

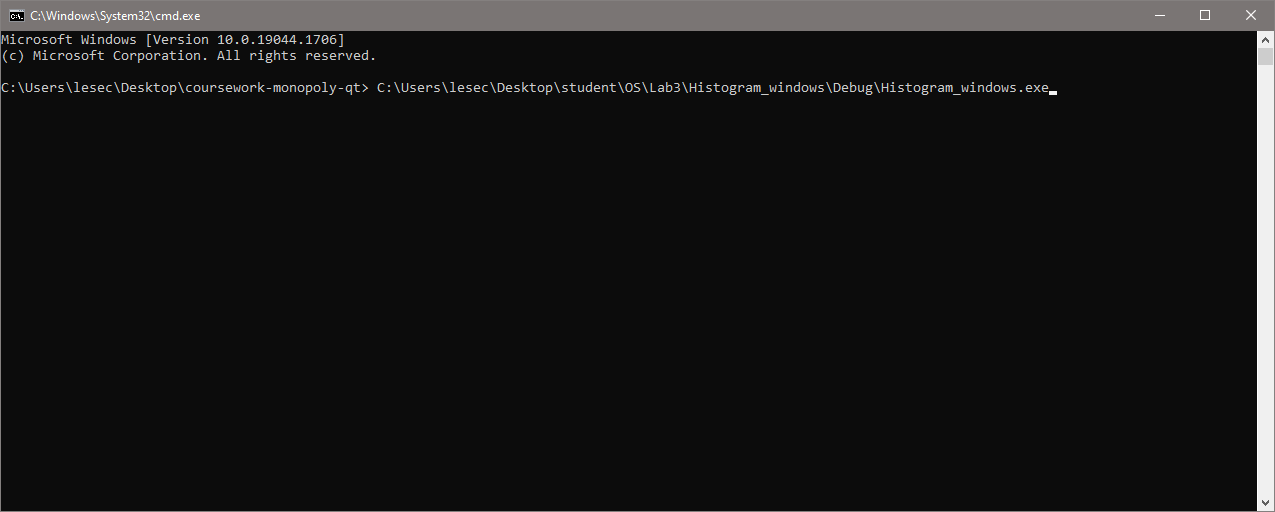


Рис. 1 Запуск програми в окремому каталозі

Після запуску ми побачимо результат.

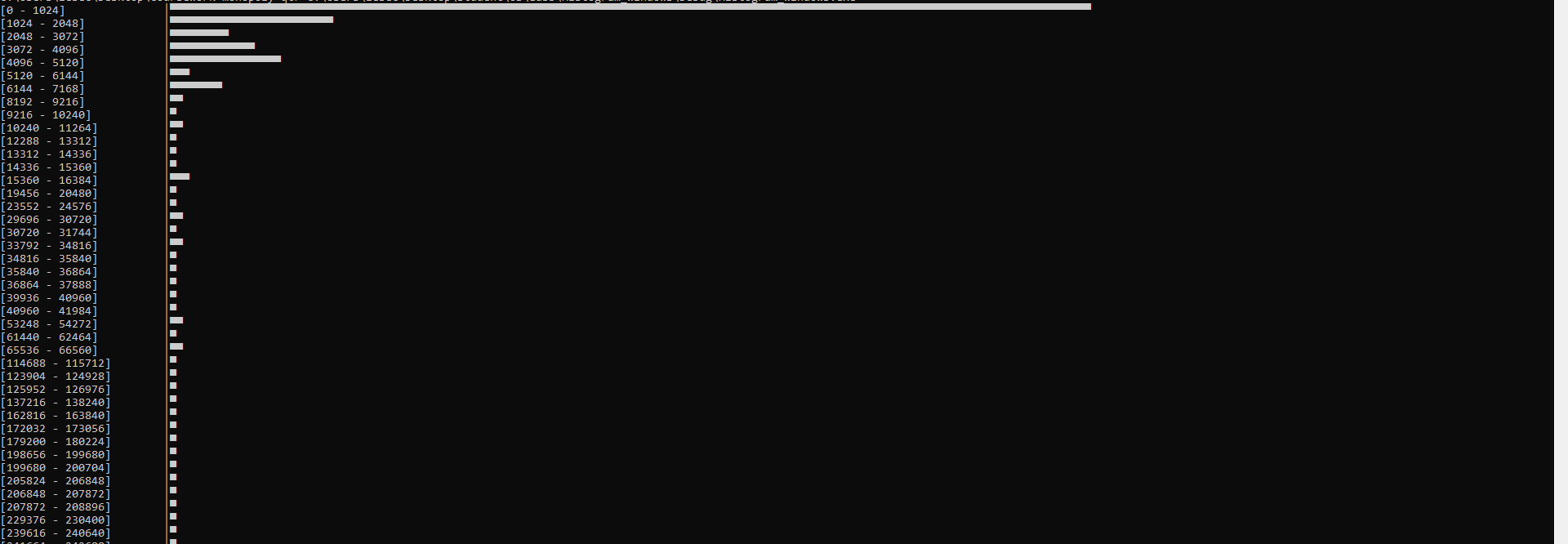


Рис. 2 Результат роботи програми

**Висновки:**

Написала утиліту, що починає роботу в заданому каталозі і спускається по дереву каталогів, записуючи по шляху розміри всіх файлів, що зустрілися їй. Закінчивши сканування каталогу, програма роздруковує гістограму розмірів файлів, використовуючи крок гістограми як параметр (наприклад, при кроці 1024, файли розміром від 0 до 1023 байт потрапляють в один інтервал, від 1024 до 2047 байт - в наступний інтервал і т.д. ).