Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №5**

По дисциплине: «Аппаратное обеспечение интеллектуальных систем»

Тема: «Разработка многопоточного приложения на основе библиотеки thread»

**Выполнил:**

Студент 2 курса

Группы ИИ-23

Макаревич Н. Р.

**Проверил:**

Михно Е.В.

Брест 2024

**Цель работы:** Изучить основные принципы работы с библиотекой thread. Получить навыки разработки многопоточных приложений на языке C++.

**Задание:**

Один поток генерирует случайные числа, второй поток записывает их в файл в той последовательности, в которой числа были сгенерированы, разбивая их на блоки по 50 значений, третий поток сортирует числа в блоках из 50 значений и записывает в файл. Размер блока должен быть варьируемый. Необходимо сгенерировать 10000 чисел

std::queue<int> numberQueue;

std::queue<int> unsortedQueue;

bool emptyGenetrator;

bool emptyRecever;

void generateNumbers() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<> dis(1, 100);

emptyGenetrator = false;

std::cout << "Check1\n";

for (int i = 0; i < 10000; ++i) {

int number = dis(gen);

numberQueue.push(number);

}

emptyGenetrator = true;

}

void writeToFile(const std::string& filename) {

std::ofstream file(filename);

std::cout << "Check2\n";

int counter = 0;

emptyRecever = false;

while (true) {

if (counter == 50) {

file << "\n";

counter = 0;

}

if (!numberQueue.empty()) {

int num = numberQueue.front();

file << num << " ";

unsortedQueue.push(num);

numberQueue.pop();

counter++;

}

if (emptyGenetrator)

break;

}

emptyRecever = true;

file.close();

}

void sortAndWriteToFile(const std::string& filename) {

std::ofstream file(filename);

std::cout << "Check3\n";

int counter = 0;

std::vector<int> numbers;

while (true) {

if (counter == 50) {

std::sort(numbers.begin(), numbers.end());

for (int i = 0; i < numbers.size(); i++)

file << numbers[i] << " ";

file << "\n";

counter = 0;

numbers = std::vector<int>();

}

if (!unsortedQueue.empty()) {

int num = unsortedQueue.front();

numbers.push\_back(num);

unsortedQueue.pop();

counter++;

}

if (emptyRecever)

break;

}

file.close();

}

int main() {

std::string filename1 = "numbers1.txt";

std::string filename2 = "numbers2.txt";

std::thread generatorThread(generateNumbers);

std::thread writerThread(writeToFile, filename1);

std::thread sorterThread(sortAndWriteToFile, filename2);

generatorThread.join();

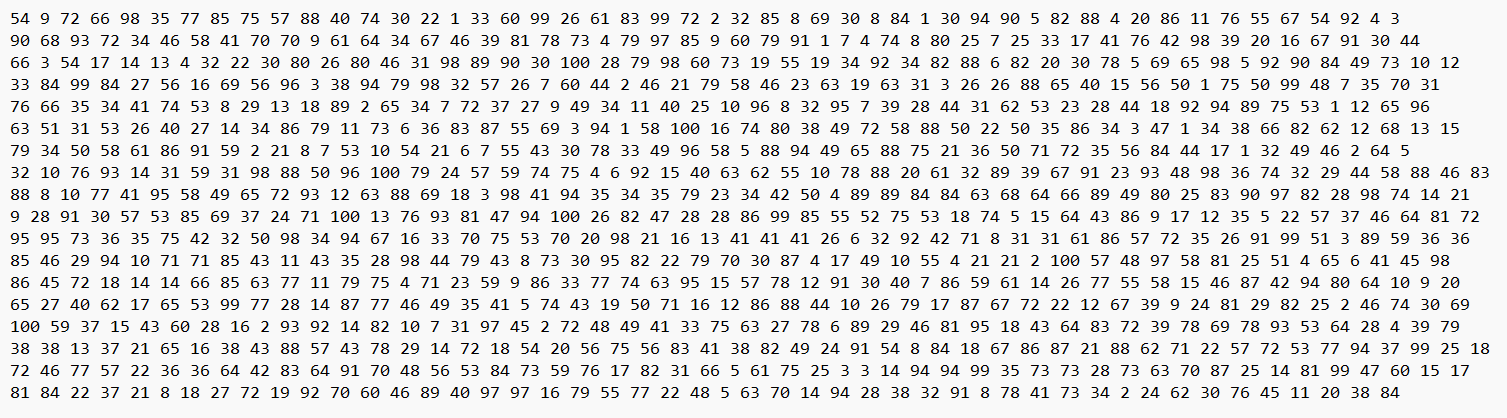
writerThread.join();

sorterThread.join();

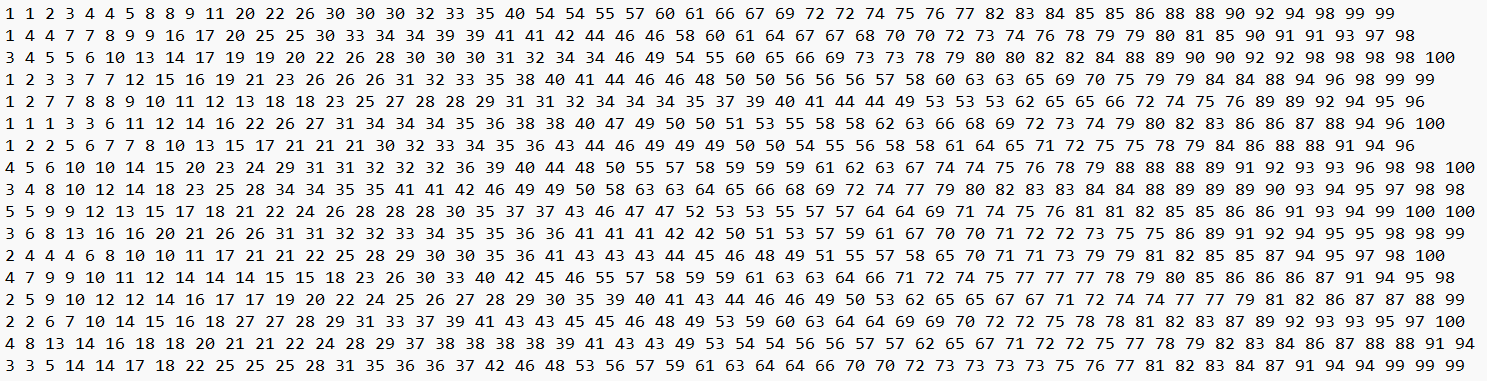
return 0;

}

Несортированные числа:



Сортированные числа:



**Вывод:** изучил основные принципы работы с библиотекой thread. Получил навыки разработки многопоточных приложений на языке с++.