

Министерство науки и высшего образования Российской Федерации  
 Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет   
имени Н. Э. Баумана   
(национальный исследовательский университет)»   
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Информационная безопасность»

ОТЧЕТ  
по лабораторной работе № 7  
по учебной дисциплине «Алгоритмические языки»  
на тему: «Изучение потоковой многозадачности»

Вариант 4

Выполнил:   
Студент 1 курса, гр. ИУ8-24  
Дудко Егор

2024

1. Цель и задачи работы:

Представить 2 варианта программы: использование шаблона future и функции async и использование класса thread***.***

2. Условие задачи:

## Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток сортирует первую половину вещественного массива, второй поток сортирует вторую половину вещественного массива, третий поток запускается после завершение первых двух, он сортирует массив полностью (уже частично отсортированный). Каждый поток имеет свое имя (например, thread1, thread2, thread3), и печатает отсортированный массив, перед печатью каждого значения элемента массива поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и массив передаются в потоковую функцию через ее параметры, использовать одну потоковую функцию для всех трех потоков. Массив предварительно до запуска потоков заполняются числами с использованием ГПСЧ. Для сортировки использовать метод прямого обмена (метод «пузырька»).

## Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.

## Между печатью имени потока и значением установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

3. Код:

Использование std::future и std::async:

#include <iostream>

#include <future>

#include <thread>

#include <random>

//Функция сортировки массива

void sortPus(const std::string& threadName, std::vector<float>& subarray, int start, int end) {

for (int i = start; i < end - 1; ++i) {

for (int j = end - 1; j > i; --j) {

if (subarray[j] < subarray[j - 1]) {

std::swap(subarray[j], subarray[j - 1]);

}

}

std::cout << threadName << ": ";

for (int k = start; k < end; ++k) {

std::cout << subarray[k] << " ";

}

std::cout << std::endl;

}

}

int main() {

//Генерация случайных чисел

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_real\_distribution<float> dis(0.0f, 20.0f);

std::vector<float> array(20);

for (float& element : array) {

element = dis(gen);

}

// Создание и запуск потоков для сортировки подмассивов

std::future<void> thread1Future = async(sortPus, "thread1", ref(array), 0, 10);

std::future<void> thread2Future = async(sortPus, "thread2", ref(array), 10, 20);

// Ожидание завершения потоков сортировки подмассивов

thread1Future.wait();

thread2Future.wait();

sortPus("thread3", ref(array), 0, 20);

std::cout << "Главный поток: все потоки закончили сортировку массива." << std::endl;

return 0;

}

Использование класса std::thread:

#include <iostream>

#include <thread>

#include <random>

void sortPus(const std::string& threadName, std::vector<float>& subarray, int start, int end) {

for (int i = start; i < end - 1; ++i) {

for (int j = end - 1; j > i; --j) {

if (subarray[j] < subarray[j - 1]) {

std::swap(subarray[j], subarray[j - 1]);

}

}

std::cout << threadName << ": ";

for (int k = start; k < end; ++k) {

std::cout << subarray[k] << " ";

}

std::cout << std::endl;

}

}

int main() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_real\_distribution<float> dis(0.0f, 20.0f);

std::vector<float> array(20);

for (float& element : array) {

element = dis(gen);

}

std::thread thread1(sortPus, "thread1", ref(array), 0, 10);

std::thread thread2(sortPus, "thread2", ref(array), 10, 20);

thread1.join();

thread2.join();

// Слияние отсортированных подмассивов в полностью отсортированный массив

sortPus("thread3", ref(array), 0, 20);

}

4. Результат программы**:**

Использование std::future и std::async:

thread1: 3.18138 15.7163 12.5993 5.83817 18.314 9.78184 14.0631 19.1457 16.1523 15.3455

thread1: 3.18138 5.83817 15.7163 12.5993 9.78184 18.314 14.0631 15.3455 19.1457 16.1523

thread1: 3.18138 5.83817 9.78184 15.7163 12.5993 14.0631 18.314 thread2: 15.3455 16.1523 19.1457

thread1: 3.18138 5.83817 9.78184 12.5993 15.7163 14.0631 15.3455 18.314 16.1523 19.1457

thread1: 3.18138 5.838172.13525 16.005 19.7542 10.5194 4.45207 12.5044 17.1564 8.67499 14.192 19.855

thread2: 2.13525 4.45207 16.005 19.7542 10.5194 8.67499 12.5044 17.1564 14.192 19.855

thread2: 2.13525 4.45207 8.67499 16.005 19.7542 10.5194 12.5044 14.192 17.1564 9.78184 12.5993 14.0631 15.7163 15.3455 16.1523 18.314 19.1457

thread1: 3.18138 5.83817 9.78184 12.5993 14.0631 15.3455 15.716319.855

thread2: 16.1523 2.13525 4.45207 8.67499 10.5194 18.31416.005 19.7542 12.5044 14.192 17.1564 19.855

thread2: 2.13525 4.45207 8.67499 10.5194 19.145712.5044 16.005 19.7542 14.192

thread1: 17.1564 3.18138 5.83817 19.855

thread2: 9.78184 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457

thread1: 3.18138 5.83817 9.78184 12.5993 14.0631 2.13525 15.3455 4.45207 15.7163 8.67499 16.1523 18.314 19.1457

thread1: 3.18138 5.83817 10.51949.78184 12.5044 14.192 16.005 19.7542 17.1564 19.855

thread2: 2.13525 4.45207 8.67499 10.5194 12.5044 14.192 16.005 12.5993 17.1564 14.0631 15.3455 15.7163 16.1523 18.314 19.1457

19.7542 19.855

thread2: 2.13525 4.45207 8.67499 10.5194 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread2: 2.13525 4.45207 8.67499 10.5194 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 5.83817 9.78184 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457 4.45207 8.67499 10.5194 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 9.78184 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457 8.67499 10.5194 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457 10.5194 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457 12.5044 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 15.3455 15.7163 16.1523 18.314 19.1457 14.192 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.1523 18.314 19.1457 16.005 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 18.314 19.1457 17.1564 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

thread3: 2.13525 3.18138 4.45207 5.83817 8.67499 9.78184 10.5194 12.5044 12.5993 14.0631 14.192 15.3455 15.7163 16.005 16.1523 17.1564 18.314 19.1457 19.7542 19.855

Главный поток: все потоки закончили сортировку массива.

Program ended with exit code: 0

Использование класса std::thread:

thread1: thread2: 0.176051 2.27428 8.21347 2.62526 3.58633 8.93268 4.10237 12.9794 0.409251 4.44283 19.4697

4.20483 thread2: 0.176051 2.27428 2.62526 8.21347 3.5863315.394 4.10237 8.93268 2.17555 12.0226 4.44283 12.9794 17.1724 15.0623 11.7051 0.799438 19.4697 18.9441

thread1:

thread2: 0.4092510.176051 2.27428 2.62526 3.58633 8.21347 4.10237 4.44283 8.93268 12.9794 19.4697

thread2: 0.176051 2.27428 2.62526 3.58633 4.10237 8.21347 4.44283 8.93268 12.9794 19.4697

thread2: 0.176051 2.27428 2.62526 3.58633 0.799438 4.20483 4.10237 4.44283 15.394 2.17555 12.0226 8.21347 8.93268 17.1724 12.9794 19.4697

thread2: 15.0623 11.7051 18.9441

0.176051 thread1: 0.409251 0.799438 2.17555 2.27428 4.20483 2.62526 3.58633 4.10237 4.44283 8.2134715.394 11.7051 12.0226 17.1724 15.0623 18.9441 8.93268 12.9794 19.4697

thread2: 0.176051 2.27428 2.62526 3.58633 4.10237 4.44283 8.21347 8.93268 12.9794 19.4697

thread2: 0.176051 2.27428 2.62526 3.58633 4.10237 4.44283

thread1: 0.409251 0.799438 2.17555 4.20483 11.7051 15.394 12.0226 15.0623 17.1724 18.9441

thread1: 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.394 15.0623 17.1724 18.9441

thread1: 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 8.21347 8.93268 12.9794 19.4697 18.9441

thread2: 0.176051 2.27428 2.62526 3.58633 4.10237

4.44283 thread1: 8.21347 8.93268 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441

thread1: 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441

thread1: 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441

12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 2.27428 2.62526 3.58633 4.10237 4.44283 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 2.62526 3.58633 4.10237 4.44283 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 3.58633 4.10237 4.44283 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 4.10237 4.44283 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 4.44283 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 8.21347 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 8.93268 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 15.0623 15.394 17.1724 18.9441 12.9794 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

thread3: 0.176051 0.409251 0.799438 2.17555 2.27428 2.62526 3.58633 4.10237 4.20483 4.44283 8.21347 8.93268 11.7051 12.0226 12.9794 15.0623 15.394 17.1724 18.9441 19.4697

Program ended with exit code: 0

5. Вывод:

В ходе выполнения лабораторной работы были изучены возможности потоковой многозадачности.