|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования«МИРЭА - Российский технологический университет»РТУ МИРЭА |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | |
| **по дисциплине** | |
| «Разработка клиент-серверных приложений» | |
| Выполнил студент группы ИКБО-01-19 | Витухина Н.А. |
|  |  |
| Принял | Зарипов Е.А. |

Практическая работа выполнена «19» сентября 2022 г.

(подпись студента)

Зачтено «\_\_» 2022 г.

(подпись студента)

Москва 2022

**Цель работы:** выполнить практические задания на тему «Примитивы синхронизации и их использование при создании клиент-серверных приложений».

**Задания:**

1. Дан массив из 10000 элементов. Необходимо написать несколько реализаций некоторой функции F в зависимости от варианта. Функция должна быть реализована следующими способами:
   1. Последовательно
   2. С использованием многопоточности (Thread, Future, и т. д.)
   3. С использованием ForkJoin.

После каждой операции с элементом массива (сравнение, сложение) добавить задержку в 1 мс при помощи Thread.sleep(1); Провести сравнительный анализ затрат по времени и памяти при запуске каждого из вариантов реализации.

Варианты функций (выбор варианта осуществляется по формуле «Номер в списке группы % 3»)

1. Поиск суммы элементов массива.
2. Поиск максимального элемента в массиве.
3. Поиск минимального элемента в массиве.
4. Программа запрашивает у пользователя на вход число. Программа имитирует обработку запроса пользователя в виде задержки от 1 до 5 секунд выводит результат: число, возведенное в квадрат. В момент выполнения запроса пользователь имеет возможность отправить новый запрос. Реализовать с использованием Future.
5. Реализовать следующую многопоточную систему. Файл. Имеет следующие характеристики:
   1. Тип файла (например XML, JSON, XLS)
   2. Размер файла — целочисленное значение от 10 до 100.

Генератор файлов -- генерирует файлы с задержкой от 100 до 1000 мс.

Очередь — получает файлы из генератора. Вместимость очереди — 5 файлов.

Обработчик файлов — получает файл из очереди. Каждый обработчик имеет параметр — тип файла, который он может обработать. Время обработки файла: «Размер файла\*7мс»

Система должна удовлетворять следующими условиям:

1. Должна быть обеспечена потокобезопасность.
2. Работа генератора не должна зависеть от работы обработчиков, и наоборот.
3. Если нет задач, то потоки не должны быть активны.
4. Если нет задач, то потоки не должны блокировать другие потоки.
5. Должна быть сохранена целостность данных.

**Описание выполнения работы:**

Обработка массива происходила согласно варианту 0 (3 в списке): Поиск суммы элементов массива. Тремя способами. Решение задачи представлено на листинге 1.

Листинг 1 – выполнение задания 1.

Результат работы программы представлен на рисунке 1.

Рисунок 1 – результат выполнения задания 1

Вторая задача заключалась в асинхронной обработке пользовательского ввода и выводе. Решение задачи представлено на листинге 2.

Листинг 2 – выполнение задания 2.

Результат работы программы представлен на рисунке 2.

Рисунок 2 – результат выполнения задания 2

В третьей задаче происходила имитация генерации файлов, сборки очереди и постепенной обработки файлов очереди. Решение задачи представлено на листингах 3-ы.

Листинг 3 – выполнение задания 3, описание.

Листинг 4 – выполнение задания 3, описание.

Листинг 5 – выполнение задания 3, описание.

Листинг 6 – выполнение задания 3, описание.

Результат работы программы представлен на рисунке 3.

Рисунок 3 – результат выполнения задания 3

**Вывод:**

В данной практической работе были отработаны навыки создания многопоточных приложений и синхронизации потоков.