Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики

Лабораторная работа №1  
по дисциплине «Параллельное программирование»

**Многопоточная программа**

|  |  |
| --- | --- |
| Факультет: | ПМИ |
| Группа: | ПМИ-91 |
| Вариант: | 8 |
| Студенты: | Гукасян Кирилл  Федяев Виктор |
| Преподаватели: | Городничев Максим Александрович  Щукин Георгий Анатольевич |

Новосибирск

2023

1. **Цель работы**

Познакомиться с базовыми возможностями многопоточного программирования; научиться работать с потоками, не имеющими информационных зависимостей.

1. **Ход работы**

Код написанных программ можно найти по ссылке: <https://github.com/somebod99/PP/tree/main/LR1>.

Имя файла соответствует пункту задания.

* 1. **Записать программу из примера 1. Скомпилировать ее и проверить корректность работы**

Программа была успешно скомпилирована и проверена на корректность работы (рисунок 1):

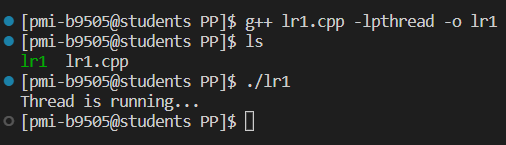


Рисунок . Результат компиляции и выполнения программы 1

* 1. **Доработать программу с целью поддержки *n* потоков**

Результат работы доработанной программы представлен на рисунке 2

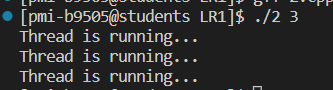


Рисунок . Результат доработанной программы

* 1. **Оценить стоимость запуска одного потока операционной системой. Изменяя количество операций, которые исполняет функция потока, определить такое их количество, чтобы порождение потока было оправданным**

Для оценки стоимости запуска одного потока операционной системой, измерим время выполнения операций внутри функции потока, а также время создания потока.

Порождение потока будет оправданным, когда время на создание потока будет меньше времени выполнения функции потока.

Результаты измерений представлены в таблице 1:

Таблица . Измерение времени

|  |  |  |
| --- | --- | --- |
| Количество операций | Время создания потока, мкс | Время выполнений функции потока, мкс |
| 100 | 59 | 1 |
| 1000 | 9 | 4 |
| 10000 | 7 | 31 |
| 100000 | 6 | 324 |
| 1000000 | 7 | 3179 |
| 10000000 | 8 | 31680 |
| 100000000 | 22 | 315311 |

Как видно из таблицы, порождение потока будет оправданным при количестве совершаемых операций внутри функции потока примерно ≥ 10000.

* 1. **Добавить в программу возможность запуска потоков с разными атрибутами**

Добавим в программу возможность запуска потока с другим минимальным размером стека.

Размер стека до внесения изменений в программу представлен на рисунке 3:

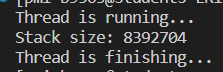


Рисунок . Размер стека до изменений

Размер стека после внесения изменений в программу представлен на рисунке 4:



Рисунок . Размер стека после изменений

* 1. **Добавить в функцию потока возможность вывода информации о всех параметрах потока, с которыми он был создан**

Выведем следующую информацию о параметрах потока:

* размер стека;
* размер защитной области;
* адрес стека;
* состояние потока.

Результат работы программы представлен на рисунке 5:

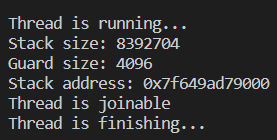


Рисунок . Вывод информации о параметрах потока

* 1. **Разработать программу, которая обеспечивает параллельное применение заданной функции к каждому элементу массива, применяемая функция и количество потоков задаются динамически**

Была разработана программа, которая обеспечивает параллельное применение заданной функции к каждому элементу массива.

Результат работа программы представлен на рисунке 6:

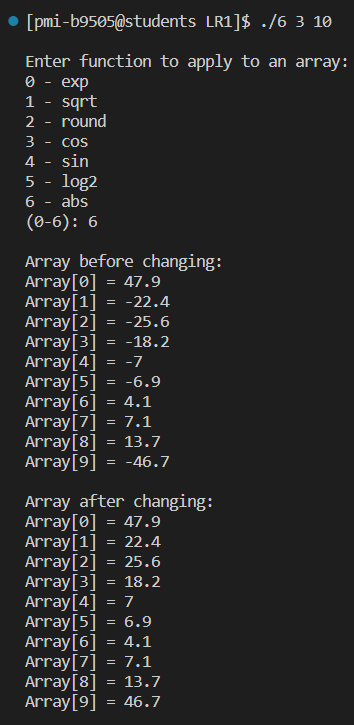


Рисунок . Применение функции к массиву

**Создать упрощенный HTTP-сервер, отвечающий на любой запрос клиента (например, браузера) строкой «Request number <номер запроса> has been processed», где под номером запроса понимается порядковый номер, присвоенный запросу сервером. Нумерация начинается с единицы**

* 1. **Доработать однопоточную версию сервера, доступную по адресу: http://rosettacode.org/wiki/Hello\_world/Web\_server#C. Обработка каждого запроса выполняется в отдельном потоке: при получении запроса создается новый поток для его обработки, после отправки результата клиенту поток завершает свою работу. Соединение с клиентом закрывается сразу после обработки запроса**

Результат работы сервера представлен на рисунках 7, 8 и 9:

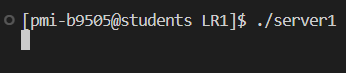


Рисунок . Работа сервера

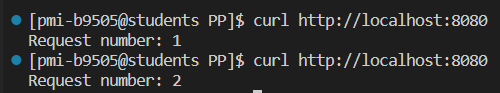


Рисунок . Отправка запроса на сервер и получение ответа

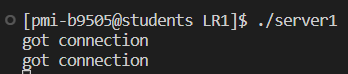


Рисунок . Работа сервера после запросов

* 1. **Оценить производительность сервера с помощью утилиты ab, входящей в комплект поставки веб-сервера Apache**

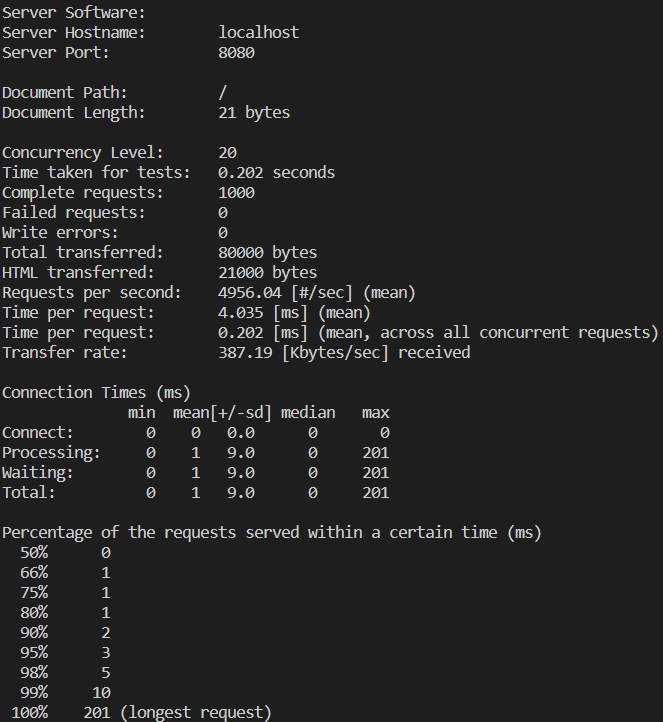


Рисунок . Оценка производительности сервера

* 1. **Оценить максимальное количество потоков, с которым может работать сервер, для различных размеров стека по умолчанию (2 Мбайт, 1 Мбайт, 512 Кбайт)**

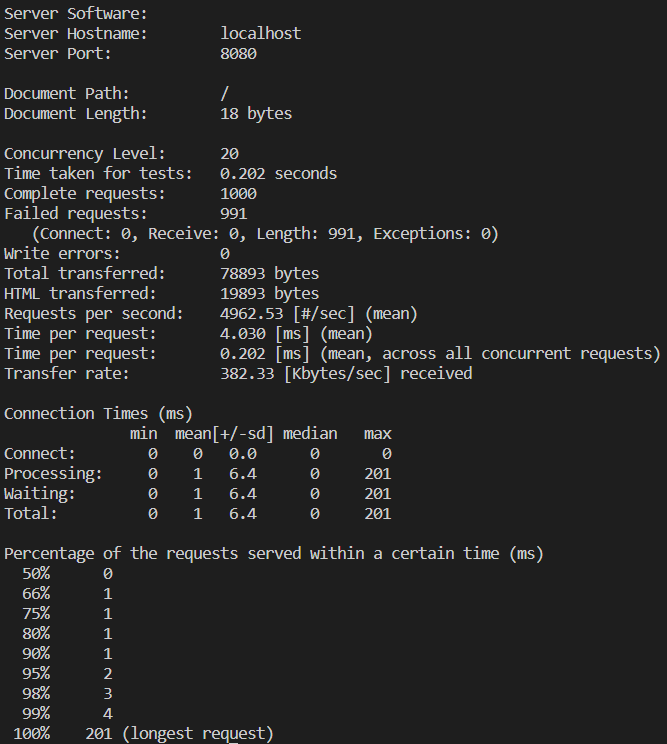
****

Рисунок . 512 Кбайт

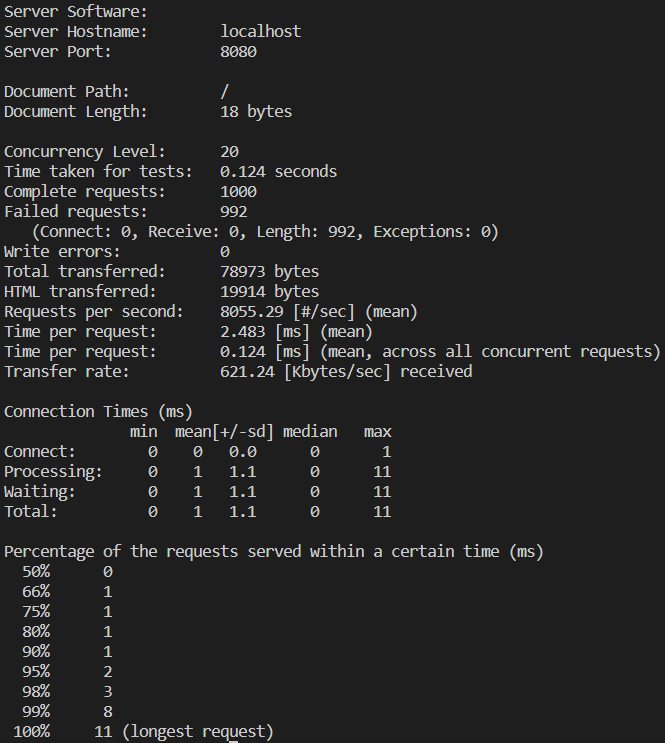
****

Рисунок . 1 Мбайт

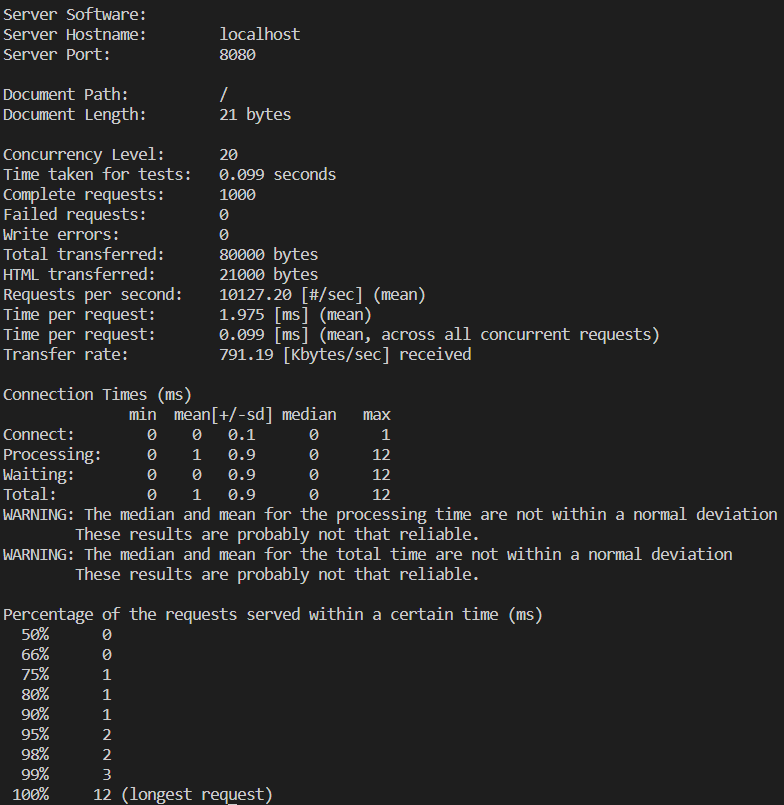
****

Рисунок . 2 Мбайт

* 1. **Добавить в обработчик запроса от клиента запуск простейшего PHP-скрипта, возвращающего версию PHP (<?php echo phpversion(); ?>). Вернуть номер версии клиенту. Оценить изменение производительности сервера с помощью утилиты ab**

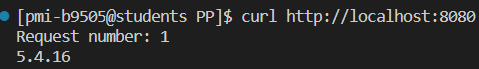
****

Рисунок . Результат работы доработанного сервера

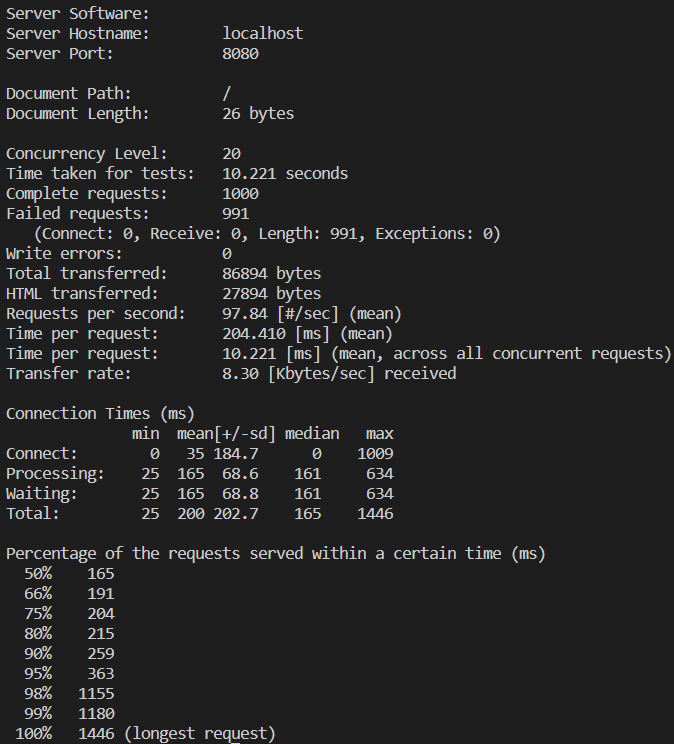


Рисунок . Оценка производительности доработанного сервера

Производительность сервера ухудшилась в результате увеличения вычислительной нагрузки.