Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

**Тема работы**

**“Потоки”**

Студент: Меджидли Махмуд Ибрагим оглы

Группа: М8О-208Б-20

Вариант: 13

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

https://github.com/loshadkaigogo/OS

**Постановка задачи**

Задача: реализовать задачу следующего рода. 128-битные числа в шестнадцатеричном представлении хранятся в файле. Необходимо вычислить их среднее арифметическое. Количество оперативной памяти подаётся.

**Общие сведения о программе**

Для реализации поставленной задачи нам нужны следующие библиотеки: <unistd.h> - для работы с системными вызовами в Linux.

<limits.h> - для определения характеристик общих типов переменных.

<stdlib.h> - для того, чтобы можно было пользоваться функциями, отвечающими за работу с памятью.

<time.h> - для функций, работающих со временем (нужно для строчки srand(time(NULL)) - для генерации рандомных чисел с использованием текущего времени).  
<pthread.h> - для работы с потоками.

<ctype.h> - для классификации и преобразования отдельных символов.

<sys/stat.h> - для доступа к файлам.

<fcntl.h> - для работы с файловым дескриптором.

<inttypes.h> - макросы для использования с функциями printf и scanf.

<string.h> - для использования функций над строками.

Для работы с потоками согласно заданию помимо библиотеки <pthreads.h> я использую такие системные вызовы, как pthread\_create, отвечающий за создание потока, имеющий тип возвращаемого значения int и принимающий 4 аргумента: указатель на поток, атрибуты по умолчанию, указатель на нужную нам функцию и аргументы), а также pthread\_join, отвечающий  
за ожидание завершения потока, имеющий тип возвращаемого значения int и принимающий 2 аргумента: указатель на поток и указатель на указатель в качестве аргумента для хранения возвращаемого значения).

Помимо системных вызовов, связанных с потоками, в моей программе имеются следующие системные вызовы:  
off\_t lseek(...) - устанавливает смещение для файлового дескриптора в значение аргумента offset.

int open(...) - открытие файлового дескриптора.

void exit(...) - выход из процесса с заданным статусом.

int close(...) - закрытие файлового дескриптора.

int write(...) - записывает количество байтов в 3 аргументе из буфера в файл с дескриптором fd.  
Программа собирается и запускается при помощи следующих команд:

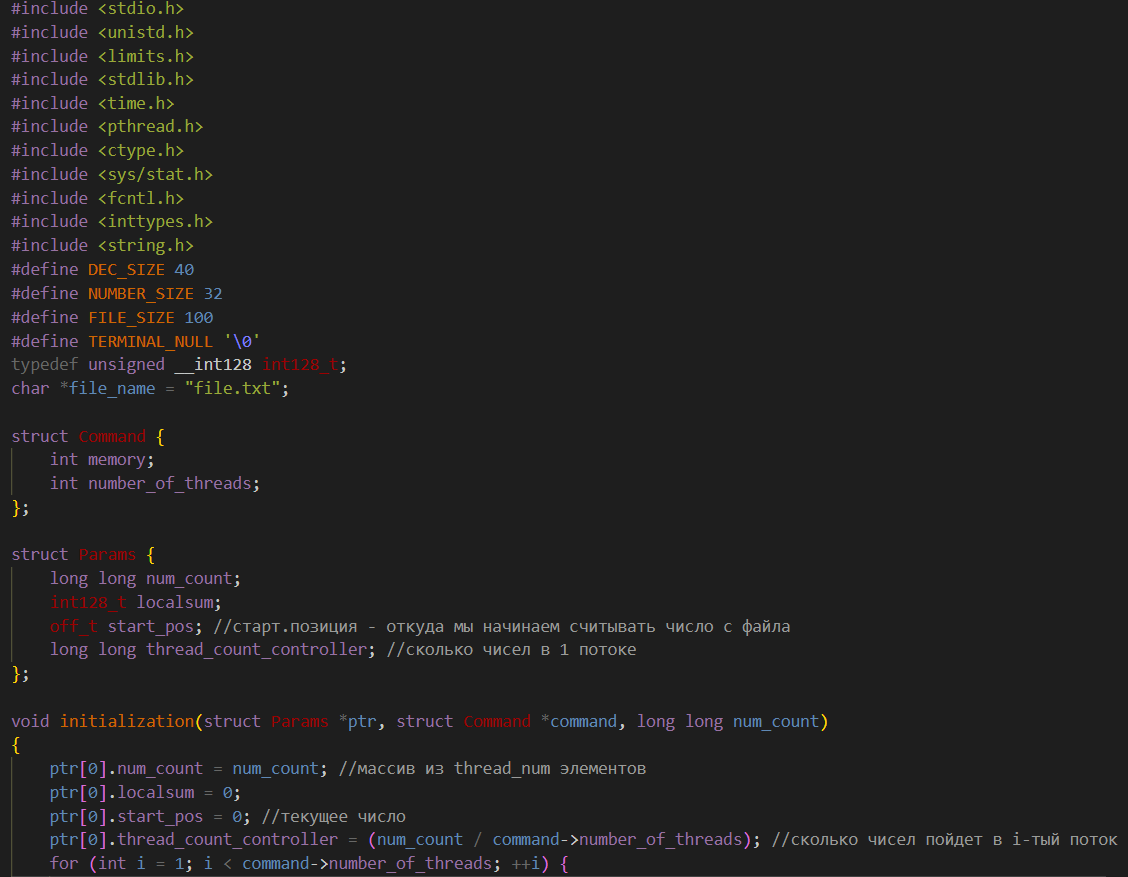
gcc lab3.c -pthread -o main

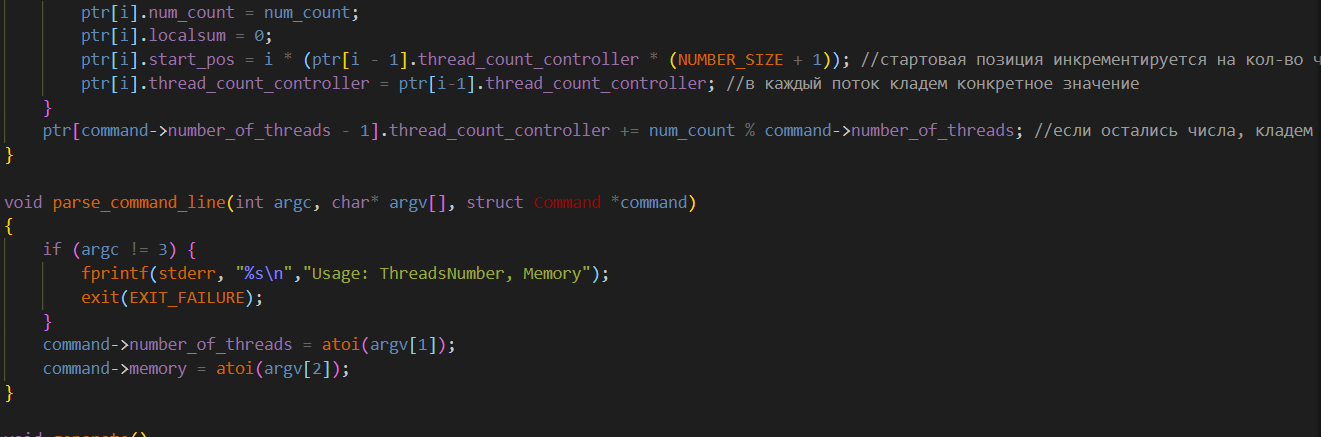
./main thread\_number memory\_amount (пример: ./main 1 72).

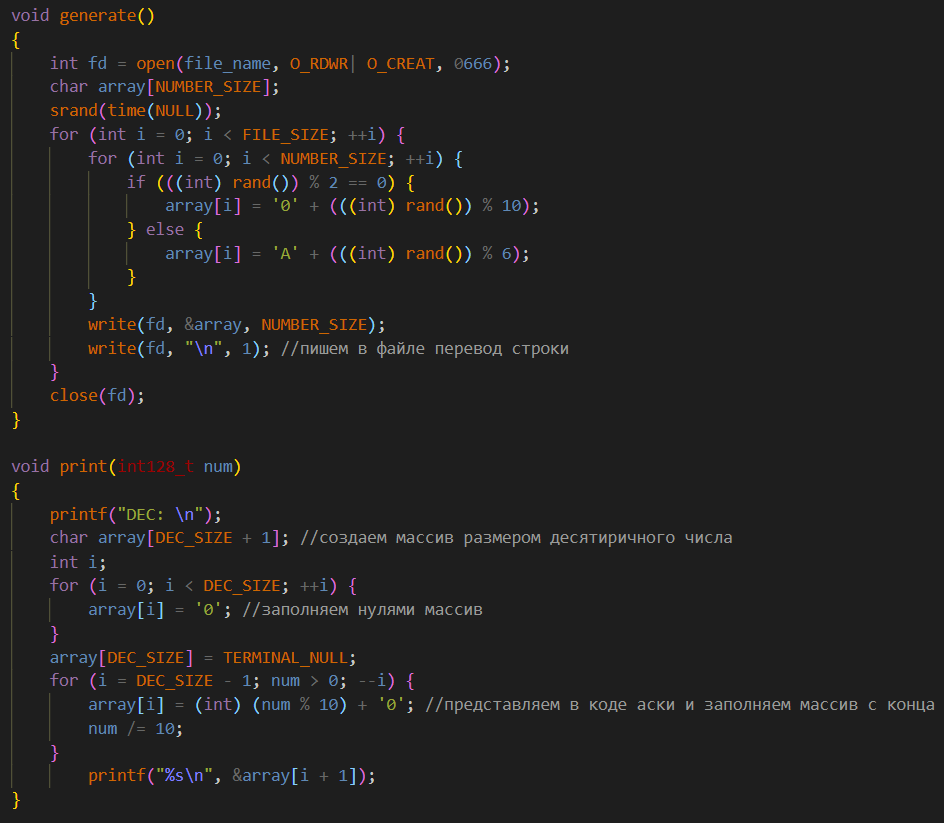
**Общий метод и алгоритм решения**

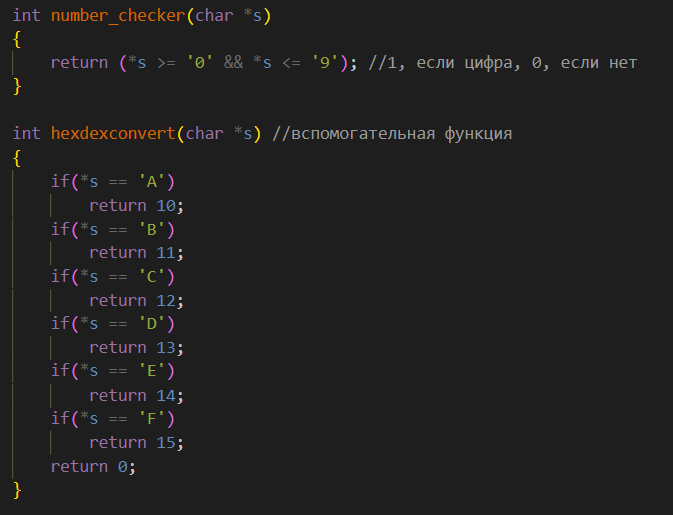
Я создаю две структуры. Первая структура struct Command отвечает за пользовательские данные: количество подающейся оперативной памяти и количество потоков. Вторая стуктура struct Params хранит в себе параметры для одного потока. С запуском программы в файле генерируются числа, и после генерации каждый поток инициализируется начальными данными в функции void initialization(...), принимающей три аргумента: указатель на пользовательскую структуру, указатель на структуру одного потока и количество чисел. Эта же функция распределяет числа по потокам. Далее каждый поток считает свою локальную сумму, в конце работы программы локальные суммы складываются и подсчитывается среднее арифметическое.

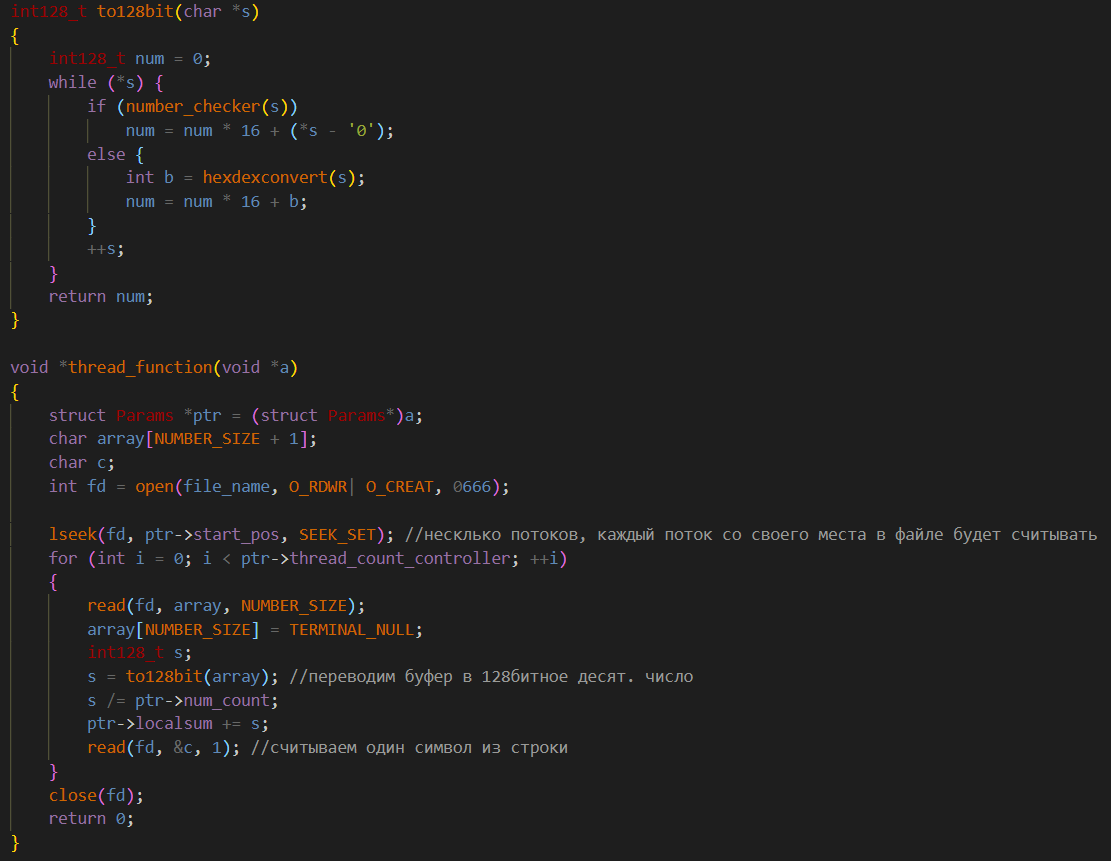
**Исходный код**

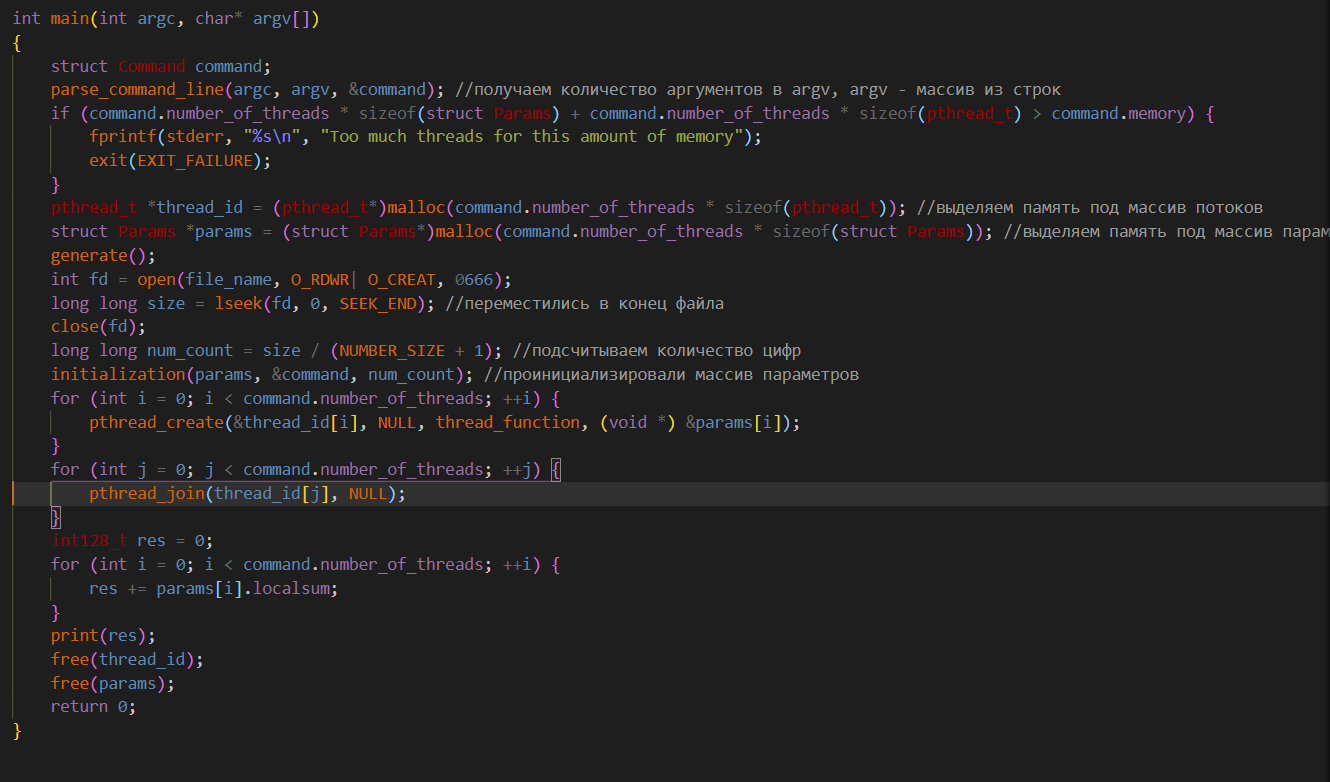
****

****

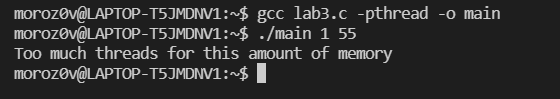
****

****

****

**  
  
  
  
Демонстрация работы программы**Тест 1.

gcc lab3.c –pthread -o main

./main 1 55 ****Тест 2.  
./main 1 56  


Тест 3.

./main 10 5600



**Выводы**Данная лабораторная работа помогла мне успешно ознакомиться с тем, как устроены потоки в Linux. Во время выполнения своего задания я изучил особенности системных вызовов и узнал многие тонкости работы с потоками.