Федеральное государственное автономное образовательное учреждение высшего образования  
 «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Дисциплина «Операционные системы»

**Лабораторная работа №2**Вариант ”FIFO”

*Выполнили*:

Студент группы P3306

Константинова Мария

*Преподаватель:*

Тюрин И.Н.

г. Санкт-Петербург

2024 год

## Задание

Для оптимизации работы с блочными устройствами в ОС существует кэш страниц с данными, которыми мы производим операции чтения и записи на диск. Такой кэш позволяет избежать высоких задержек при повторном доступе к данным, так как операция будет выполнена с данными в RAM, а не на диске (вспомним пирамиду памяти).

В данной лабораторной работе необходимо реализовать блочный кэш в пространстве пользователя в виде динамической библиотеки (dll или so). Политику вытеснения страниц и другие элементы задания необходимо получить у преподавателя.

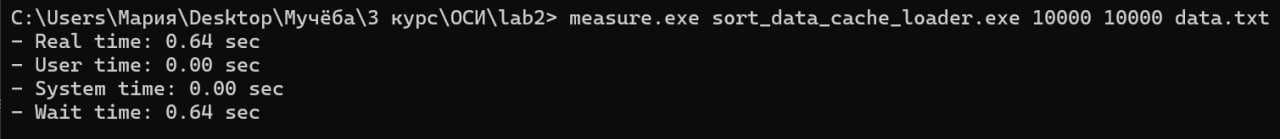
При выполнении работы необходимо реализовать простой API для работы с файлами, предоставляющий пользователю следующие возможности:

1. Открытие файла по заданному пути файла, доступного для чтения. Процедура возвращает некоторый хэндл на файл. Пример:  
   int lab2\_open(const char \*path).
2. Закрытие файла по хэндлу. Пример:  
   int lab2\_close(int fd).
3. Чтение данных из файла. Пример:  
   ssize\_t lab2\_read(int fd, void buf[.count], size\_t count).
4. Запись данных в файл. Пример:  
   ssize\_t lab2\_write(int fd, const void buf[.count], size\_t count).
5. Перестановка позиции указателя на данные файла. Достаточно поддержать только абсолютные координаты. Пример:  
   ​​​​​​​off\_t lab2\_lseek(int fd, off\_t offset, int whence).
6. Синхронизация данных из кэша с диском. Пример:  
   int lab2\_fsync(int fd).

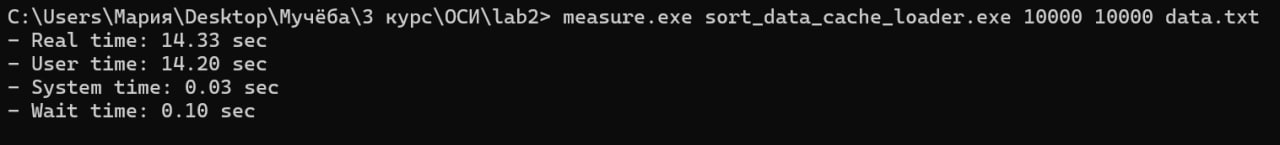
## Краткий обзор кода

## Данные о работе программы-нагрузчика до и после внедрения

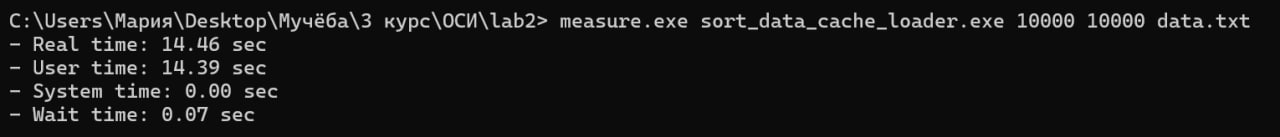
#### С реализованным кэшем



С не отключенным системным кэшем



С отключенным системным кэшем



## Заключение

Наиболее эффективным оказался вариант с реализованным кэшем, чтение и запись выполняются быстрее благодаря предварительному хранению блоков в памяти и минимизации количества обращений к диску.

Механизмы операционной системы не всегда эффективно кэшируют и управляют файлами, особенно при частых чтениях и записях небольшими блоками.

Отключение системного кэша (FILE\_FLAG\_NO\_BUFFERING) также дало плохие результаты, каждое обращение к файлу идёт напрямую на диск без предварительного кэширования.