**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Лабораторная работа № 4**

Тема: Процессы, file mapping

Студент: Гамов Павел

Группа: 80-207

Преподаватель: Миронов Е.С.

Дата:

Оценка:

Москва, 2019

**1.**               **Постановка задачи**

В отличие от второй лабораторной работы, где необходимо было использовать pipe для связи между процессами, в этой лабораторной работе необходимо использовать общую память, системные вызовы mmap, munmap, fork. Задача реализовать систему взаимодействия между процессами в стиле клиент-сервер, где на стороне сервера выполняется проверка числа на простоту, вызывая каждый новый уровень рекурсии в новом процессе. Для этого использовать систему mutex, futex, sem, cond\_var.

**2.**           **Описание программы**

В главной функции сразу вызывается раздвоение программы на сервер и клиент. Для последовательной работы были использованы два мьютекса, когда клиент проходил цикл, получал число, он отпускал мьютекс и его ловил сервер, сервер обрабатывал число, которое хранится во внешней глобальной переменной, стракте, содержащей массив чисел, в том числе необходимых для хранения состояния сервера и тд.

Когда сервер получает число, он рекурсивно вызывает функцию prime, которая строит “башню” из процессов, после чего получает ответ и сворачивает постройку обратно до двух процессов, записывая результат в общий массив данных.

**3.**           **Выводы**

Использование общей памяти позволяет ускорить написание подобных задач, потому что нагромождение пайпов и их обработка занимает свое место, читабельность кода уменьшается. Отображение на память стало для меня новинкой, как новый маллок, только в разы интереснее, так как к нему имеют доступ все дочерние процессы.

Но в свою очередь возникает проблема доступа к ресурсам, а также проблема очередности. Все это решается теми или иными способами с помощью mutex и sem концепций.