Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Студент: Царев Михаил Вадимович

Группа: М8О-208Б-21

Вариант: 12

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**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/de1phin/OS/lab4

**Постановка задачи**

**Цель работы**

Приобретение практических навыков в:

* Освоение принципов работы с файловыми системами
* Обеспечение обмена данных между процессами посредством технологии «File mapping»

**Задание**

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или через отображаемые файлы (memory-mapped files).

Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Лабораторная работа №2:

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или каналы (pipe). Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Блок вариантов 5: Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод.

12 вариант) Child1 переводит строки в верхний регистр. Child2 убирает все задвоенные пробелы.

**Общие сведения о программе**

Программа состоит из двух исходных файлов – parent.c и child.c. Parent создает временный файл, для которого вызывает системный вызов mmap, который, вместе с сигналами, используется для общения между Parent, Child1 и Child2.

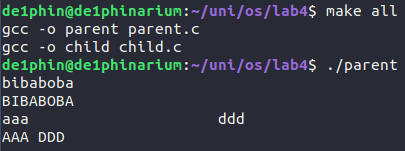
**Общий метод и алгоритм решения**

Parent создает временный файл, резервирует в нем BUF\_SIZE память, и отображает её с помощью mmap и запускает Child1, передавая ему созданный файл. Child1 порождает Child2 и они оба вызывают для переданного им файла mmap. Parent ждет ввода, пока Child1 и Child2 «спят» в бесконечном цикле, передает им сигнал SIGUSR1, по которому Child1 читает и обрабатывает строчку, точно так же посылая сигнал Child2, который в свою очередь делает свою часть работы и посылает SIGUSR2 Parent, для того чтобы тот прочитал результат, вывел его пользователю, и снова стал читать ввод

**Исходный код**

В репозитории.

**Демонстрация работы программы**



**Выводы**

Создал систему процессов, обрабатывающих данные и коммуницирующих при помощи системных вызовов mmap, signal и kill.