Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

Студент: Муханов Андрей Ильич

Группа: М8О-209Б-23

Вариант: 12

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Шаги по изменению лабораторной номер 1
4. Демонстрация работы программы
5. Выводы

**Репозиторий**

https://github.com/andreymuch/OC\_MAI

**Постановка задачи**

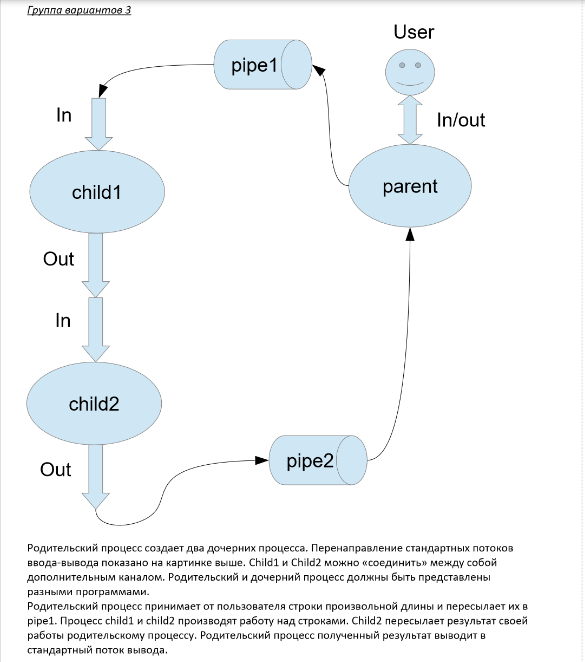
**Цель работы**

Приобретение практических навыков в:

1. Освоение принципов работы с файловыми системами
2. 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.

****

****

**Шаги по изменению лабораторной номер 1**

Шаг 1. Анализ начального кода с использованием pipe

* Исходный проект был построен на механизме именованных каналов (Named Pipes) для обмена данными между процессами.
* Родительский процесс создавал два именованных канала: Pipe1 и Pipe2. Первый использовался для отправки данных дочернему процессу 1, второй – для получения данных от дочернего процесса 2.
* В child1 данные преобразовывались в верхний регистр, а затем передавались в Pipe3 для обработки в child2.
* child2 удалял лишние пробелы из строки и передавал результат обратно в Pipe2.

Шаг 2. Переход на File Mapping

* Проект был переписан для использования отображаемого файла, который обеспечивает общую память между процессами.
* Родительский процесс создает отображаемый файл с помощью CreateFileMapping и отображает его в память через MapViewOfFile.
* Дочерние процессы подключаются к этой общей памяти, используя OpenFileMapping и MapViewOfFile.

Шаг 3. Организация ожидания завершения child1 в child2

* В общей памяти был выделен специальный участок для передачи дескриптора процесса child1.
* После завершения работы child1, child2 считывает данные из памяти и продолжает выполнение.

Шаг 4. Реализация кода

1. Родительский процесс:
   * Создание отображаемого файла для хранения данных и передачи дескриптора процесса child1.
   * Запуск дочерних процессов.
   * Передача строки для обработки в общую память.
   * Ожидание завершения работы child2.
2. Дочерний процесс 1 (child1):
   * Подключение к общей памяти.
   * Считывание строки, преобразование в верхний регистр.
   * Запись результата обратно в общую память.
3. Дочерний процесс 2 (child2):
   * Подключение к общей памяти.
   * Ожидание завершения child1.
   * Удаление лишних пробелов в строке.
   * Запись финального результата в общую память.

Результаты

* Проект успешно переписан на использование File Mapping.
* Осуществлена синхронизация между процессами: child2 корректно ожидает завершения работы child1.
* Все операции с общей памятью реализованы безопасно и эффективно.
* Код протестирован и работает согласно заданию.

**Демонстрация работы программы**

**Введите строку: hgjfdhgjkfhdkj hgjkfdhkjghjkfdhkjghd hgjfhgjfjgfj ghjfgjfhgjhf**

**Результат: HGJFDHGJKFHDKJ HGJKFDHKJGHJKFDHKJGHD HGJFHGJFJGFJ GHJFGJFHGJHF**

**Выводы**

1. **Освоение технологий межпроцессного взаимодействия**  
   В ходе выполнения работы был изучен и применен механизм отображаемых файлов (File Mapping) для обмена данными между процессами. Это позволило обеспечить безопасный и эффективный способ передачи данных между основным процессом и дочерними.
2. **Реализация синхронизации процессов**  
   Успешно реализована синхронизация между дочерними процессами. Дочерний процесс 2 (child2) корректно ожидает завершения работы дочернего процесса 1 (child1), что исключает возможные конфликты при доступе к общей памяти.
3. **Переход от Named Pipes к File Mapping**  
   Переписывание проекта с использования именованных каналов (Named Pipes) на отображаемые файлы значительно упростило архитектуру программы и улучшило ее производительность за счет прямого доступа к общей памяти.
4. **Обработка ошибок**  
   Реализована обработка ошибок, возникающих при работе с отображаемыми файлами, процессами и системными функциями. Это повышает надежность программы и облегчает ее отладку.
5. **Реализация задания**  
   Программа полностью соответствует требованиям задания: создаются два дочерних процесса, между которыми обеспечивается обмен данными через общую память. Также реализовано последовательное преобразование строки: сначала в верхний регистр (child1), затем удаление лишних пробелов (child2).
6. **Практическое значение работы**  
   Полученные знания и навыки могут быть применены для создания сложных систем с межпроцессным взаимодействием, таких как клиент-серверные приложения или программы обработки данных в многопоточном окружении.

Работа выполнена в полном соответствии с заданием, протестирована, и продемонстрировала корректную работу всех механизмов.