

Министерство образования Республики Беларусь

**Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

**Отчёт по лабораторной работе №2
по курсу «ОС» на тему:
«Алгоритмы синхронизации процессов»**

Выполнил студент группы 921703:

Кравцов Михаил Сергеевич

Проверил:

Садовский Михаил Ефимович

Минск 2021

Содержание:

1. Описание задачи
2. Описание методов
3. Описание алгоритма
4. Схем алгоритма
5. Вывод

1. Описание задачи

Вариант: 3.

Задача - Написать программу, копирующую файл. Программа должна запускать два дочерних процесса, один читает файл, другой пишет. Передача данных между процессами должна быть реализована через общую область памяти, синхронизация – с помощью объектов ядра. Материнский процесс должен обрабатывать ошибки в дочерних процессах (например, если при записи произошла ошибка – читающий процесс должен завершаться материнским).

2. Описание методов

Для решения задачи были применены:

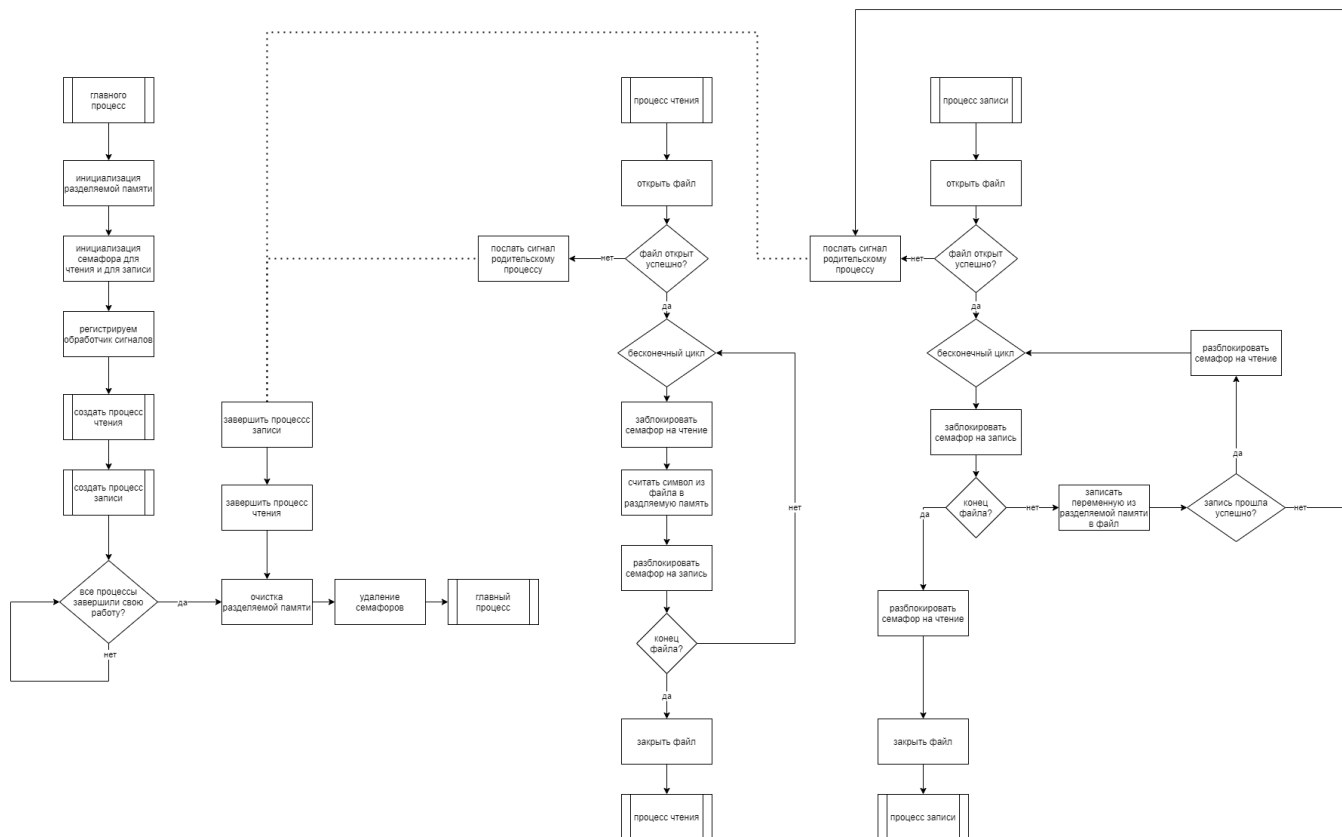
- общая память (ftok, shmget, shmat, shmdt, shmctl)
- семафоры (sem_t, sem_open, sem_close, sem_unlink)
- сигналы (signal, kill, pause)

Для корректной работы процесса чтения и процесса записи данных в файл были использованы методы `sem_post` и `sem_wait`. Также была использована разделяемая память для сохранения результатов операции чтения из файла, а позже для операции записи в файл. Для обработки материнским процессом ошибок, которые могут возникнуть в дочерних процессах, был применен механизм сигналов.

3. Описание алгоритма

Родительский процесс создает два дочерних процесса – один для чтения из файла, другой для записи в файл. Дочерние процессы общаются друг с другом через семафоры. Процесс чтения посимвольно записывает данные из файла в разделяемую память. Процесс чтения посимвольно записывает в файл данные из разделяемой памяти. Когда процесс для чтения считал символ и записал его в разделяемую память он информирует об этом процесс чтения и блокируется. Процесс чтения записывает символ из разделяемой памяти в файл, после чего информирует процесс чтения и блокируется. Если возникает какая-то ошибка, то процесс чтения/записи посылает сигнал родительскому процессу. Главные процесс ждет окончания работы дочерних процессов и после очищает разделяемую память и семафоры. Главные процесс содержит обработчик сигнала, который может поступить от дочерних процессов при возникновении ошибок. Обработчик завершает оба процесса (на запись и на чтение).

4. Схемы алгоритма



5. Вывод

В результате выполнения лабораторной работы я познакомился с примитивами синхронизации в операционной, а также разделяемой памятью в системе Linux, а также реализовал потокобезопасное приложение с общими данными.

В процессе реализации я изучил проблему синхронизации потоков и сложности в создании потокобезопасных программ. Получил практические знания о семафорах, сигналах, разделяемой памяти в операционной системе Linux.