Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №3 по курсу

«Операционные системы»

Тема работы

“Межпроцессорное взаимодействие через memory-mapping files”

Студент: Мазепа Илья Алексеевич

Группа: М8О-209Б-23

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2024

GitHub репозиторий: https://github.com/Tyhyqo/mai_oc

Цель работы

Приобретение практических навыков в:

– Освоении принципов работы с файловыми системами

– Обеспечении 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) должна создать для решения 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.

Вариант

Родительский процесс создает два дочерних процесса. Перенаправление стандартных потоков ввода-вывода показано на картинке выше. Child1 и Child2 можно «соединить» между собой дополнительным каналом. Родительский и дочерний процесс должны быть представлены разными программами.

Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1. Процесс child1 и child2 производят работу над строками. Child2 пересылает результат своей работы родительскому процессу. Родительский процесс полученный результат выводит в стандартный поток вывода.

Child1 переводит строки в верхний регистр. Child2 превращает все пробельные символы в символ «_».

Основные технологии и моменты

Освоение принципов работы с файловыми системами

- Использование системного вызова shm_open() для создания и открытия отображаемых файлов.
- Использование системного вызова mmap() для отображения файлов в память.
- Использование системного вызова munmap() для освобождения отображаемой памяти.
- Использование системного вызова shm_unlink() для удаления отображаемых файлов.

Обеспечение обмена данных между процессами

- Использование семафоров (sem_open(), sem_wait(), sem_post(), sem_close(), sem_unlink()) для синхронизации процессов.
- Создание и инициализация семафоров с помощью sem_open() и sem_init().
- Ожидание на семафоре с помощью sem_wait().
- Освобождение семафора с помощью sem_post().
- Закрытие и удаление семафоров с помощью sem_close() и sem_unlink().

Обработка системных ошибок

- Проверка ошибок при создании и открытии отображаемых файлов.
- Проверка ошибок при отображении и освобождении памяти.
- Проверка ошибок при создании и использовании семафоров.

Пример работы программы

1. Родительский процесс принимает строку от пользователя.
2. Строка записывается в отображаемый файл и сигнализируется первому дочернему процессу (child1) через семафор.
3. Child1 преобразует строку в верхний регистр и сигнализирует второму дочернему процессу (child2) через семафор.

4. Child2 заменяет все пробельные символы на символ «_» и сигнализирует родительскому процессу через семафор.

5. Родительский процесс читает преобразованную строку из отображаемого файла и выводит результат на экран.

Вывод

В результате выполнения лабораторной работы были приобретены практические навыки в освоении принципов работы с файловыми системами и обеспечении обмена данных между процессами посредством технологии «File mapping». Программа успешно создает два дочерних процесса, которые обрабатывают строки, переданные родительским процессом, и возвращают результат обратно родительскому процессу для вывода.