# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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»

Студент: Власко М. М.
Группа: М8О-208Б-23
Преподаватель: Живалев Е. А.
Дата:
Оценка:
Полпись:

Тема: Работа с файловыми системами и технологиями отображения памяти в ОС

Цель работы: Приобретение практических навыков в:

- Освоении принципов работы с файловыми системами.
- Обеспечении обмена данных между процессами посредством технологии "File mapping" (отображаемые файлы).

**Вариант:** 21. Правило фильтрации: нечетные строки отправляются в pipe1, четные в pipe2. Дочерние процессы инвертируют строки.

#### Задачи:

- 1. Разработать программу на языке Си, реализующую работу с процессами и их взаимодействие через отображаемые файлы (memory-mapped files).
- 2. Обеспечить взаимодействие между процессами с использованием системных сигналов/событий.
- 3. Обрабатывать системные ошибки, которые могут возникнуть в результате работы программы.
- 4. Реализовать инвертирование строк в дочерних процессах и записывать результат в файлы, заданные пользователем.

#### Описание решения:

Программное решение состоит из трех основных компонентов:

## 1. Родительский процесс (Parent):

- о Инициализирует именованные семафоры и отображаемую память.
- о Создает два дочерних процесса.
- Осуществляет чтение строк из потока ввода и передает их в отображаемую память.
- о Использует семафоры для управления доступом к памяти.

## 2. Дочерние процессы (Child1 и Child2):

- о Получают доступ к отображаемой памяти и семафорам.
- Читают строки, инвертируют их с помощью функции ReverseString, и записывают результат в файлы, заданные пользователем.
- о Завершают выполнение при получении специального сигнала "q".

#### 3. Вспомогательные модули (utils.c):

о Содержат реализацию вспомогательных функций, таких как ReverseString.

# Логика работы программы:

- 1. Родительский процесс создает именованные семафоры и отображаемую память с помощью shm open и mmap.
- 2. Пользователь вводит имена двух файлов, которые передаются в дочерние процессы в качестве аргументов.
- 3. Родительский процесс создает два дочерних процесса с помощью fork и запускает их с помощью execl.
- 4. Родительский процесс:
  - о Принимает строки от пользователя.
  - о Помещает строки в отображаемую память.
  - о Сигнализирует дочерним процессам через семафоры о доступности данных.
- 5. Дочерние процессы:
  - о Читают строки из отображаемой памяти.
  - о Инвертируют строки и записывают их в файлы.
  - о Возвращают управление родительскому процессу через семафоры.
- 6. Программа завершает работу при вводе специального символа "q".

**Репозиторий:** https://github.com/MMVlasko/mai-os-lab-work/tree/main/LW3

Исходный код: Программное обеспечение состоит из следующих файлов:

- 1. **main.c:** Основная функция, которая вызывает родительский процесс.
- 2. **parent.c:** Логика работы родительского процесса (инициализация ресурсов, управление дочерними процессами).
- 3. **child1.c и child2.c:** Логика работы дочерних процессов (обработка строк и запись в файлы).
- 4. utils.c: Реализация вспомогательных функций, включая ReverseString.

## Пример функции ReverseString:

```
void ReverseString(char *str) {
  int len = strlen(str);
  for (int i = 0; i < len / 2; i++) {
    char temp = str[i];
    str[i] = str[len - i - 1];</pre>
```

```
str[len - i - 1] = temp;
}
```

# Пример работы:

```
./lab3

first
second

lol
gone

q
// first
lol
// second
```

**Вывод:** В ходе выполнения лабораторной работы были выполнены все поставленные задачи. Программа успешно организует взаимодействие между процессами через отображаемую память и семафоры. Функциональность инвертирования строк и записи в файлы реализована корректно. Были приобретены практические навыки работы с отображаемыми файлами, семафорами и обработкой ошибок. Программа протестирована на операционной системе Linux и показала стабильную работу.