

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: Григорьев Т. А.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2024

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: 18. Правило фильтрации: нечетные строки отправляются в pipe1, четные в pipe2. Дочерние процессы удаляют все гласные из строк.

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 удаление гласных в дочерних процессах и записывать результат в файлы, заданные пользователем.

Описание решения:

Программное решение состоит из трех основных компонентов:

1. **Родительский процесс (Parent):**
 - Инициализирует именованные семафоры и отображаемую память.
 - Создает два дочерних процесса.
 - Осуществляет чтение строк из потока ввода и передает их в отображаемую память.
 - Использует семафоры для управления доступом к памяти.
2. **Дочерние процессы (Child1 и Child2):**
 - Получают доступ к отображаемой памяти и семафорам.
 - Читают строки, удаляют в них гласные с помощью функции `RemoveVowels`, и записывают результат в файлы, заданные пользователем.
 - Завершают выполнение при получении специального сигнала "q".
3. **Вспомогательные модули (utils.c):**

- Содержат реализацию вспомогательных функций, таких как `RemoveVowels`.

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 `exec1`.
4. Родительский процесс:
 - Принимает строки от пользователя.
 - Помещает строки в отображаемую память.
 - Сигнализирует дочерним процессам через семафоры о доступности данных.
5. Дочерние процессы:
 - Читают строки из отображаемой памяти.
 - Удаляют в них гласные и записывают их в файлы.
 - Возвращают управление родительскому процессу через семафоры.
6. Программа завершает работу при вводе специального символа "q".

Репозиторий: <https://github.com/timofeez/os-labs/tree/main/03>

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 `RemoveVowels`.

Пример функции `RemoveVowels`:

```
void RemoveVowels(char* str) {  
  
    char* src = str;  
  
    char* dst = str;
```

```

while (*src) {

    char lower = tolower((unsigned char)*src);

    if (lower != 'a' && lower != 'e' && lower != 'i' && lower != 'o' &&
lower != 'u') {

        *dst++ = *src;

    }

    src++;

}

*dst = '\\0';

```

} Пример работы:

```

./lab3

first

second

lol

gone

q

// first

ll

// second

gn

```

Вывод: В рамках выполнения лабораторной работы все поставленные задачи были успешно выполнены. Разработанная программа эффективно организует взаимодействие между процессами с использованием отображаемой памяти и семафоров. Реализация функционала удаления гласных из строк и записи в файлы выполнена корректно. В процессе работы были приобретены практические навыки работы с отображаемыми файлами, семафорами и обработки ошибок. Программа прошла тестирование на операционной системе Linux и продемонстрировала стабильность в работе.