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 №1

По курсу «Операционные системы»

Студент: Лернер Ф. Л.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

**Тема:** Управление процессами и межпроцессное взаимодействие в ОС

**Цель работы:** Приобретение практических навыков в:

* Управлении процессами в операционной системе.
* Организации обмена данными между процессами посредством каналов (pipe).

**Вариант:** 12. Child1 переводит строки в верхний регистр. Child2 убирает все задвоенные пробелы

**Задачи:**

1. Разработать программу на языке Си, реализующую управление процессами и их взаимодействие через каналы.
2. Реализовать фильтрацию строк, вводимых пользователем, для передачи их в соответствующие процессы через каналы.
3. Обеспечить обработку ошибок, возникающих при выполнении программы.
4. Выполнить инвертирование строк в дочерних процессах и записать результат в файл.

**Описание решения:** Программное решение состоит из трех основных модулей:

* **Parent (родительский процесс):** создает два дочерних процесса и организует межпроцессное взаимодействие через каналы.
* **Child1 и Child2 (дочерние процессы):** принимают строки из соответствующих каналов, инвертируют их и записывают результат в указанные пользователем файлы.

Программа функционирует следующим образом:

1. Родительский процесс создает два канала и выполняет два вызова fork для создания дочерних процессов.
2. После создания каналов пользователь вводит два имени файлов, которые передаются в качестве аргументов дочерним процессам.
3. Родительский процесс принимает строки произвольной длины от пользователя. Child1 переводит строки в верхний регистр, Child2 убирает пробелы
4. Дочерние процессы (Child1 и Child2):
   * Читают строки из соответствующего канала.
   * Выполняют инвертирование строки.
   * Записывают инвертированные строки в указанные файлы.
5. Обработка ошибок предусмотрена на всех этапах работы программы (например, ошибки открытия файлов, создания процессов, записи и чтения данных).

**Исходный код:** Программное обеспечение состоит из следующих файлов:

* **CMakeLists.txt**
* **child1.cpp**
* **child1.hpp**
* **child2.cpp**
* **child2.hpp**
* **main.cpp**
* **parent.cpp**
* **parent.hpp**
* **test\_utils.cpp**
* **utils.cpp**
* **utils.hpp**

**Репозиторийhttps://github.com/LernerF/labs\_os**

**Пример кода:**

void startChild1(int pipe1[], int pipe3[]) {

close(pipe1[1]);

close(pipe3[0]);

char buffer[BUFFER\_SIZE];

ssize\_t n;

while ((n = read(pipe1[0], buffer, BUFFER\_SIZE)) > 0) {

buffer[n] = '\0';

toUpperCase(buffer);

write(pipe3[1], buffer, strlen(buffer) + 1);

**Вывод:** В ходе выполнения лабораторной работы были выполнены все поставленные задачи. Программа успешно создает два дочерних процесса и организует обмен данными между ними и родительским процессом посредством каналов. Фильтрация строк, их инвертирование и запись в файлы выполняются корректно. Были приобретены практические навыки в работе с процессами, каналами и обработке ошибок в операционных системах. Программа протестирована на операционной системе Linux и показала стабильную работу.