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: Татаркин И. В.

Группа: М8О-208Б-23

Преподаватель: Живалев И. В.

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:** 1. Пользователь вводит команды вида: «число число число<endline>». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс считает их сумму и выводит её в файл. Числа имеют тип int. Количество чисел может быть произвольным.

**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 суммирование строк в дочерним процессе и записать результат в файл.

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

* **Parent (родительский процесс):** создает дочерний процесс и организует межпроцессное взаимодействие через канал.
* **Child:** принимает числа из канала, суммирует их и записывает результат в указанный пользователем файл.

Программа функционирует следующим образом:

1. Родительский процесс создает канал и выполняет вызов fork для создания дочернего процесса.
2. После создания канала пользователь вводит имя файла, который передается в качестве аргумента дочернему процессу.  
      
    Дочерний процесс   
   * Читает числа из соответствующего канала.
   * Выполняет суммирование чисел.
   * Записывает сумму чисел в указанный файл.

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

1. **main.c**: Инициализация родительского процесса.
2. **parent.c**: Основная логика родительского процесса (создание каналов, управление дочерним процессом).
3. **child.c** : Логика дочернего процесса (суммирование чисел и запись в файл).
4. **utils.c**: Вспомогательная функция для суммирования чисел.

**Репозиторий:** https://github.com/IvanTatarkin/OS-Labs/tree/main/LW1

**Пример кода:**

// Пример функции calculste\_sum из utils.c

int calculate\_sum(const char \*input) {

int sum = 0;

int num;

char \*input\_copy = strdup(input);

if (input\_copy == NULL) {

return 0;

}

char \*token = strtok(input\_copy, " ");

while (token != NULL) {

num = atoi(token);

sum += num;

token = strtok(NULL, " ");

}

free(input\_copy);

return sum;

}

**Пример работы:**

./lab1

"Введите имя файла: "

test.txt

"Введите числа через пробел (закончите ввод нажатием Enter): "

5 6 7 8

9 -4 0

end

// test.txt

26

5

**Вывод:** В процессе выполнения лабораторной работы ознакомились с процессами, их видами, состояниями, научились передавать данные между процессами через каналы, ознакомились с системными вызовами и научились их применять на практике.