

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу
«Операционные системы»**

**Тема работы
«Использование mmap и семафоров»**

Студент: Лютоев Илья Александрович
Группа: М8О-207Б-21
Вариант: 6
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

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) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или mmap. Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Группа вариантов №2

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия файла с таким именем на чтение. Стандартный поток ввода дочернего процесса переопределяется открытым файлом. Дочерний процесс читает команды из стандартного потока ввода. Стандартный поток вывода дочернего процесса перенаправляется в pipe1. Родительский процесс читает из pipe1 и прочитанное выводит в свой стандартный поток вывода. Родительский и дочерний процесс должны быть представлены разными программами.

Вариант 6

В файле записаны команды вида: «число число число». Дочерний процесс считает их сумму и выводит результат в стандартный поток вывода. Числа имеют тип int. Количество чисел может быть произвольным. img

Методы и алгоритмы решения

Откроем файл введенный пользователем и создаем pipe. Используем fork() и execlp() для создания дочернего процесса child, в него мы передаем наш pipe и файл. После этого с помощью dup2() перенаправляем файл в поток ввода, а поток вывода в переданный pipe, после чего считываем данные из файла и обрабатываем их. После чего в parent мы перенаправляем pipe в поток вывода.

Выполнение программы

/home/lyutoev/workshop/os/OS/lab4/cmake-build-debug/main

Input file path: ../1

6
12
21
0
3
73
10
26
46

Process finished with exit code 0

Входной файл

1 2 3
3 4 5
6 7 8
0 0 0
1 1 1
33 5 35
10 0 0
1 22 3
12 12 22

Вывод

В ходе выполнения работы я узнал о работе с системными вызовами, процессами и mmap.