

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**ОТЧЕТ
О ВЫПОЛНЕНИИ 1 ЛАБОРАТОРНОЙ РАБОТЫ
ПО ДИСЦИПЛИНЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ»**

Выполнил(а) студент

группы М8О-208Б-23

Романов Вадим Михайлович

Проверили и приняли:

Живалев Е.А.

Катаев Ю. И.

Москва, 2024

Тема: «Первая программа»

Цели работы:

Приобретение практических навыков в:

- Управление процессами в ОС
- Обеспечение обмена данных между процессами посредством каналов

Задание (вариант №4):

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса пишет имя файла, которое будет передано при создании дочернего процесса. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс передает команды пользователя через `pipe1`, который связан с стандартным входным потоком дочернего процесса. Дочерний процесс при необходимости передает данные в родительский процесс через `pipe2`. Результаты своей работы дочерний процесс пишет в созданный им файл.

Пользователь вводит команды вида: «число число число». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс производит деление `parent child(args: fileName) pipe1 pipe2 In/out User File In Out` первого числа, на последующие, а результат выводит в файл. Если происходит деление на 0, то тогда дочерний и родительский процесс завершают свою работу. Проверка деления на 0 должна осуществляться на стороне дочернего процесса. Числа имеют тип `float`. Количество чисел может быть произвольным.

Идея, метод, алгоритм решения задачи:

Работа программы начинается с функции `haupt` (тот же `main`, только в профиль). Там принимается на вход строка с числами, помещается в массив, и с помощью

функции `process_input_from_pipe` данные с родительского процесса передаются в дочерний. Дочерний же процесс с помощью функции `process_input_from_pipe` производит вычисления и записывает ответ в файл.

Вывод:

Работа позволила освоить базовые принципы многопроцессного программирования, работу с каналами межпроцессного взаимодействия и систему сборки CMake. В процессе программирования была одержана первая победа над линкером CMake, который очень не хотел запускать какие-либо программы. Во всяком случае, основы для будущего многопоточного программирования заложены.