Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

**Тема работы**

Студент: Попов Матвей Романович

Группа: М8О-208Б-20

Вариант: 1

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

<https://github.com/.../os_lab2>

**Постановка задачи**

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 со 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. Допускается просто открыть файл и писать туда, не перенаправляя стандартный поток вывода. 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.

**Общие сведения о программе**

Программа представляет из себя один файл main.cpp.

**Общий метод и алгоритм решения**

С помощью вызова fork создаются родительский и дочерний процессы, родительский процесс считывает название будущего файла и некоторое количество векторов целых чисел, которые передаются в дочерний процесс. Дочерний процесс создаёт файл и записывает в него сумму чисел из каждого вектора. Ключей для запуска программа не имеет.

**Исходный код**

#include <unistd.h>

#include <iostream>

#include <vector>

#include <string>

#include <fstream>

using namespace std;

int main()

{

    int fd[2];

    pipe(fd);

    int id = fork();

    if (id == -1)

    {

        return -1;

    }

    if (id == 0)

    {

        string filename;

        int length;

        read(fd[0], &length, sizeof(int));

        for (int i = 0; i < length; i++)

        {

            char c;

            read(fd[0], &c, sizeof(char));

            filename.push\_back(c);

        }

        ofstream outfile(filename);

        int t;

        read(fd[0], &t, sizeof(int));

        int amount;

        read(fd[0], &amount, sizeof(int));

        int sum = 0;

        for (int i = 0; i < amount; i++)

        {

            int n;

            read(fd[0], &n, sizeof(int));

            sum += n;

        }

        for (int i = 0; i < t; i++)

        {

            int amount;

            read(fd[0], &amount, sizeof(int));

            int sum = 0;

            for (int i = 0; i < amount; i++)

            {

                int n;

                read(fd[0], &n, sizeof(int));

                sum += n;

            }

            outfile << sum << endl;

        }

        outfile.close();

        close(fd[0]);

        close(fd[1]);

    }

    else

    {

        cout << "Parent's PID: " << getpid() << endl;

        cout << "Child's PID: " << id << endl;

        vector<int> numbers;

        string filename;

        cout << "Enter the file name:\n";

        cin >> filename;

        int length = filename.length();

        write(fd[1], &length, sizeof(int));

        for (int i = 0; i < length; i++)

        {

            write(fd[1], &filename[i], sizeof(char));

        }

        cout << "Enter amount of commands:\n";

        int t;

        cin >> t;

        t++;

        write(fd[1], &t, sizeof(int));

        for (int i = 0; i < t; i++)

        {

            string s;

            getline(cin, s);

            vector<int> numbers;

            string n;

            for (int i = 0; i < s.length(); i++)

            {

                if ((s[i] != ' ')||(s[i] != '\0'))

                {

                    n.push\_back(s[i]);

                }

                if ((s[i] == ' ')||(s[i] == '\0')||(s[i] == '\n'))

                {

                    int num = stoi(n);

                    n = "";

                    numbers.push\_back(num);

                }

            }

            int amount = numbers.size();

            write(fd[1], &amount, sizeof(int));

            for (int i = 0; i < amount; i++)

            {

                write(fd[1], &numbers[i], sizeof(int));

            }

        }

        close(fd[1]);

        close(fd[0]);

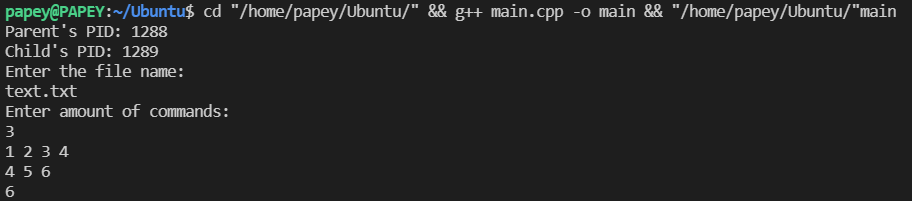
    }

    return 0;

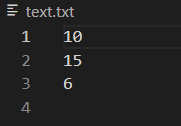
}

**Демонстрация работы программы**

Ввод в консоль:



Содержимое файла text.txt:



**Выводы**

Проделав лабораторную работу, я приобрёл практические навыки в управлении процессами в ОС и обеспечил обмен данными между процессами посредством каналов.