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: Фаттяхетдинов Сильвестр Динарович

Группа: М8О-208Б-20

Вариант: 17

Преподаватель:

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/silverfatt/OS/tree/main/Lab2

**Постановка задачи**

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами.

Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод.

Правило фильтрации: строки длины больше 10 символов отправляются в pipe2, иначе в pipe1. Дочерние процессы удаляют все гласные из строк.

**Общие сведения о программе**

Вся программа содержится в одном файле main.cc

**Общий метод и алгоритм решения**

Запуск осуществляется при помощи ввода в командную строку unix:

./main

**Исходный код**

Добавьте исходный код вашей программы (для вставки кода в Microsoft Word можно взять способ [здесь](https://stackoverflow.com/questions/387453/how-do-you-display-code-snippets-in-ms-word-preserving-format-and-syntax-highlig)).

main.cc

#include "unistd.h"

#include <iostream>

#include <string>

#include <fstream>

int main(){

    std::string fname1, fname2;

    std::cout << "Enter names of files: " << std::endl;

    std::cin >> fname1;

    std::cin >> fname2;

    std::fstream fos;

    int fd1[2];

    pipe(fd1);

    int fd2[2];

    pipe(fd2);

    if (pipe(fd1) == -1) {

        std:: cout << "Pipe1 error!" << std:: endl;

        return 1;

    }

    if (pipe(fd2) == -1) {

        std:: cout << "Pipe2 error!" << std:: endl;

        return 1;

    }

    int cid1 = fork();

    if (cid1 == -1)

    {

        std:: cout << "Fork error!" << std:: endl;

        return -1;

    }

    else if (cid1 == 0) // --------------------[CHILD1]--------------------

    {

        fos.open(fname1, std::fstream::in | std::fstream::out | std::fstream::app);

        int N;

        read(fd1[0], &N, sizeof(int));

        while(N>0){

        int size;

        read(fd1[0], &size, sizeof(int));

        char strc[size];

        read(fd1[0], strc, sizeof(char) \* size);

        std::string str;

        for (int i = 0; i < size; i++){

            str.push\_back(strc[i]);

        };

        int j = 0;

        while (j < str.size()) {

            while ((str[j] == 'a') or (str[j] == 'e') or (str[j] == 'i') or (str[j] == 'o') or (str[j] == 'u') or (str[j] == 'y') or

                (str[j] == 'A') or (str[j] == 'E') or (str[j] == 'I') or (str[j] == 'O') or (str[j] == 'U') or (str[j] == 'Y'))

            {

                str.erase(str.begin() + j);

            }

                j++;

            };

        fos << str << std::endl;

        std::cout << "[CHILD1] " << str << std:: endl;

        N--;

        };

        close(fd1[0]);

        close(fd1[1]);

    }// --------------------[CHILD1]--------------------

    else

    {

        int cid2 = fork();

        if (cid2 == -1)

        {

        std:: cout << "Fork error!" << std:: endl;

        return -1;

        }

        else if (cid2 == 0) {// --------------------[CHILD2]--------------------

            fos.open(fname2, std::fstream::in | std::fstream::out | std::fstream::app);

            int N;

            read(fd2[0], &N, sizeof(int));

            while(N>0){

            int size;

            read(fd2[0], &size, sizeof(int));

            char strc[size];

            read(fd2[0], strc, sizeof(char) \* size);

            std::string str;

            for (int i = 0; i < size; i++){

                str.push\_back(strc[i]);

            };

            int j = 0;

            while (j < str.size()) {

                while ((str[j] == 'a') or (str[j] == 'e') or (str[j] == 'i') or (str[j] == 'o') or (str[j] == 'u') or (str[j] == 'y') or

                    (str[j] == 'A') or (str[j] == 'E') or (str[j] == 'I') or (str[j] == 'O') or (str[j] == 'U') or (str[j] == 'Y'))

                {

                    str.erase(str.begin() + j);

                }

                    j++;

            }

            fos << str << std:: endl;

            std::cout << "[CHILD2] " << str << std:: endl;

            N--;

            };

        close(fd2[0]);

        close(fd2[1]);

        }// --------------------[CHILD2]--------------------

        else {// --------------------[PARENT]--------------------

        int N;

        std::cout << "[PARENT] Enter number of strings" << std::endl;

        std::cin >> N;

        write(fd1[1], &N, sizeof(int));

        write(fd2[1], &N, sizeof(int));

        std::cout <<"[PARENT] Enter " << N << " string(s): " << std::endl;

        for (int i = 0; i < N; i++){

            std::string str1;

            std::cin >> str1;

            int k = str1.size();

            char strc1[k];

            for (int i = 0; i < k; i++){

                strc1[i] = str1[i];

            }

            if (str1.size() <= 10) {

                write(fd1[1], &k, sizeof(int));

                write(fd1[1], strc1, sizeof(char) \* k);

            }

            else{

                write(fd2[1], &k, sizeof(int));

                write(fd2[1], strc1, sizeof(char) \* k);

            }

        }

                close(fd1[0]);

                close(fd1[1]);

                close(fd2[0]);

                close(fd2[1]);

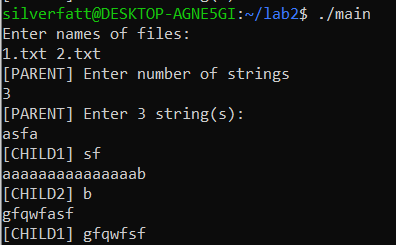
        }// --------------------[PARENT]--------------------

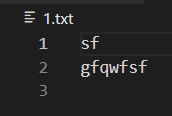
    }

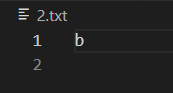
    return 0;

}

**Демонстрация работы программы**







**Выводы**

Я приобрёл навыки в управлении процессами в ОС Unix и обеспечении обмена данных между процессами при помощи каналов.