Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Студент: Пирогов М.Д.

Группа: М8О-207Б-21

Вариант: 2

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**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/DKazhekin/OС

**Постановка задачи**

**Цель работы**

Приобретение практических навыков в:

1. Освоение принципов работы с файловыми системами
2. Обеспечение обмена данных между процессами посредством технологии «File mapping»

**Задание**

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 float. Количество чисел может быть произвольным.

**Общие сведения о программе**

Программа компилируется из файла parent.c.

**Общий метод и алгоритм решения**

В самом начале в родительском процессе нам необходимо создать файл data.txt, куда мы запишем название файла, в который выведется конечный ответ, его длина (файла), количество чисел, введенных пользователем и сами числа. Далее мы «мапим» наш data.txt в память с помощью mmap()

В дочернем процессе мы уже обращаемся к той области памяти, где находится наш файл и считываем все данные, вычисляя ответ и записывая его в файл с названием, которое указал пользователь.

Исходный код

#include "unistd.h"

#include "stdio.h"

#include <stdlib.h>

#include "string.h"

#include <sys/mman.h>

#include "sys/stat.h"

#include "sys/types.h"

#include "sys/wait.h"

#include "errno.h"

#include "fcntl.h"

int main() {

int fd = open("data.txt", O\_RDONLY);

struct stat sb;

if (fstat(fd, &sb) == -1){

perror("Couldn't get the file size\n");

}

int id = fork();

if (id < 0) {

perror("An error occurred with fork");

return -1;

} else if (id == 0) { // Сhild

sleep(15);

char \*out = static\_cast<char\*>(mmap(NULL, sb.st\_size, PROT\_READ, MAP\_SHARED, fd, 0));

printf("%p Child\n", out);

int nOfChars = out[0] - '0';

int nOfDigits = out[nOfChars + 3] - '0';

char str[nOfChars];

for (int i = 2; i < 2 + nOfChars; i++) {

str[i - 2] = out[i];

}

float sum = 0;

int start = 5 + nOfChars;

for (int j = 0; j < nOfDigits; ++j) {

int i = 0;

char a[4];

while (i < 4) {

a[i] = out[start];

++i;

++start;

}

sum += atof(a);

start += 1;

}

FILE \*fptr2;

fptr2 = fopen(str, "w");

fprintf(fptr2,"The answer is: %.2f\n", sum);

fclose(fptr2);

close(fd);

} else { // Parent

FILE \*fp;

fp = fopen("data.txt", "w");

char s[30];

printf("Enter the filename:\n");

scanf("%s", s);

int len = strlen(s);

int n = rand() % 9 + 1;

fprintf(fp, "%d\n", len);

fprintf(fp, "%s\n", s);

fprintf(fp, "%d\n", n);

printf("Enter %d different digits\n", n);

float x;

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

scanf("%f", &x);

fprintf(fp, "%.2f\n", x);

}

fclose(fp);

char \*in = static\_cast<char\*>(mmap(NULL, sb.st\_size, PROT\_READ, MAP\_SHARED, fd, 0));

printf("%p Parent\n", in);

close(fd);

pid\_t wpid;

int status = 0;

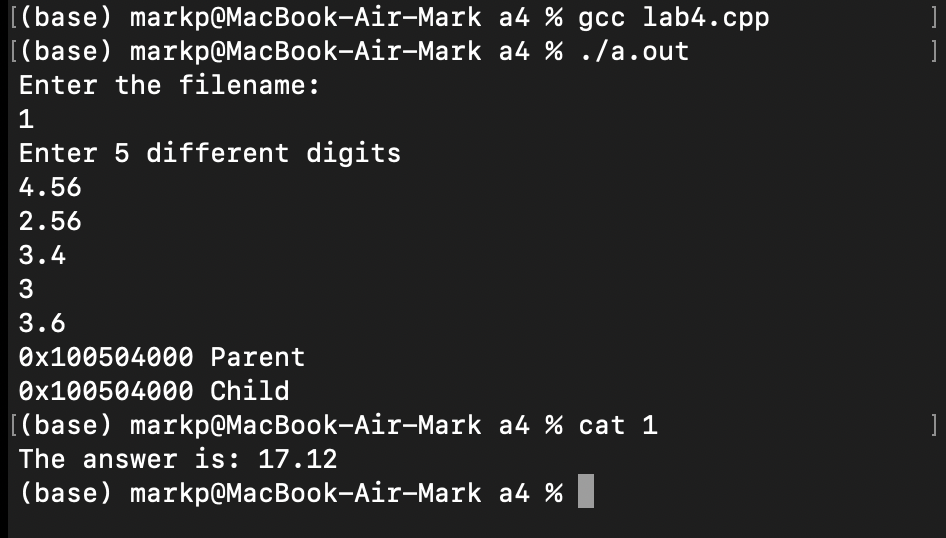
while ((wpid = wait(&status)) > 0);

}

return 0;

}

**Демонстрация работы программы**

Здесь мы видим, что указатель на область памяти в дочернем процессе такой же, как и у родителя

В файле с ответом наблюдаем верные вычисления

**Выводы**

Составлена и отлажена программа на языке Си, осуществляющая работу и взаимодействие между процессами с использованием отображаемых файлов. Так, получены навыки в обеспечении обмена данных между процессами посредством MMF.