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: Казанцев Данила Игоревич
Группа: М8О-207Б-21
Вариант: 3
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Содержание

Репозиторий

https://github.com/thgdanilaya/mai_os_labs

Постановка задачи

Цель работы

Приобретение практических навыков в:

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

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 через отображаемые файлы (memory-mapped files).

Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Общие сведения о программе

Программа компилируется из файла main.cpp

Общий метод и алгоритм решения

Создал структуру asd, чтобы хранить полученные от пользователя данные и операцию дочернего процесса. Создал для структуры отображенную память, доступную для обоих процессов, а для регулировки доступа процессов к памяти использую семафор.

Исходный код

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <unistd.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <semaphore.h>

using namespace std;

int human_get(sem_t *semaphore) {
    int s;
    sem_getvalue(semaphore, &s);
```

```
    return s;
}

void human_set(sem_t *semaphore, int n) {
    while (human_get(semaphore) < n) {
        sem_post(semaphore);
    }
    while (human_get(semaphore) > n) {
        sem_wait(semaphore);
    }
}

struct abc {
    int num;
    int st;
};

int main() {
    int ans = 0;
    abc *mapped = (abc *) mmap(0, sizeof(abc), PROT_READ | PROT_WRITE,
MAP_SHARED | MAP_ANONYMOUS, 0, 0);
    if (mapped == MAP_FAILED) {
        cout << "mmap error\n";
        return -1;
    }
    sem_unlink("_sem");
    sem_t *sem = sem_open("_sem", O_CREAT, 0, 2);
    string filename;
    int n;
    ofstream out;
    cout << "Enter name of the file:\n";
    getline(cin, filename);
    cout << "Enter some numbers:\n";
    int id = fork();
    if (id < 0) {
        cout << "fork error\n";
        return -1;
    }
    if (id == 0) {
        out.open(filename);
        while (true) {
            while (human_get(sem) == 2) {
                continue;
            }
            if (mapped->st == 1) {
                if (mapped->num == 0) {
                    cout << "Division by zero\n";
                    out << "Division by zero" << endl;
                    exit(1);
                } else {
                    ans /= mapped->num;
                }
                out << ans << endl;
                ans = 0;
                human_set(sem, 2);
            } else if (mapped->st == 2) {
                if (mapped->num == 0) {
                    cout << "Division by zero\n";
                    out << "Division by zero" << endl;
                    exit(1);
                } else {
                    ans /= mapped->num;
                }
```

```
                out << ans << endl;
                out.close();
                human_set(sem, 0);
                exit(0);
            } else if (mapped->st == 0) {
                if (ans == 0) {
                    ans = mapped->num;
                } else if (mapped->num == 0) {
                    cout << "Division by zero\n";
                    out << "Division by zero" << endl;
                    exit(1);
                } else {
                    ans /= mapped->num;
                }
                human_set(sem, 2);
            }
        }
    } else if (id > 0) {
        while (human_get(sem) != 0) {
            char c;
            scanf("%d%c", &n, &c);
            mapped->num = n;
            if (c == ' ') {
                mapped->st = 0;
            }
            if (c == '\n') {
                mapped->st = 1;
            }
            if (c == '\0') {
                mapped->st = 2;
            }
            human_set(sem, 1);
            while (human_get(sem) == 1) {
                continue;
            }
        }
    }
    munmap(mapped, sizeof(abc));
    sem_close(sem);
    sem_destroy(sem);
    return 0;
}
```

Демонстрация работы программы

```
danilaya@DESKTOP-JFEGEK0:/mnt/c/Users/frede/CLionProjects/oslab4$ g++ -pthread main.cpp -o main
danilaya@DESKTOP-JFEGEK0:/mnt/c/Users/frede/CLionProjects/oslab4$ ./main
Enter name of the file:
out
Enter some numbers:
6 3
9 3 3
9 0
Division by zero
^Z
[4]+  Stopped                 ./main
danilaya@DESKTOP-JFEGEK0:/mnt/c/Users/frede/CLionProjects/oslab4$ cat out
2
1
Division by zero
danilaya@DESKTOP-JFEGEK0:/mnt/c/Users/frede/CLionProjects/oslab4$
```

Выводы

Я приобрел практические навыки, необходимые для работы с отображаемой памятью и семафорами.