Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №0 по курсу**

**«Операционные системы»**

**Тема работы**

**“Освоение технологии «File mapping»”**

Студент: Зинин Владислав Владимирович

Группа: М8О-208Б-20

Вариант:5

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/frankeloff/OS

**Постановка задачи**

Задача: 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. Результаты своей работы дочерний процесс пишет в созданный им файл. Пользователь вводит команды вида: «число». Далее это число передается от родительского процесса в дочерний. Дочерний процесс производит проверку на простоту. Если число составное, то в это число записывается в файл. Если число отрицательное или простое, то тогда дочерний и родительский процессы завершаются. Обеспечить обмен данных между процессами посредством технологии «File mapping»

**Общие сведения о программе**

int id = fork () - создание дочернего процесса, в переменной id будет лежать “специальный код” процесса (-1 - ошибка fork, 0 - дочерний процесс, >0 - родительский);

void \* mmap(void \**start*, size\_t *length*, int *prot* , int *flags*, int *fd*, off\_t *offset*) где Функция mmap отражает *length* байтов, начиная со смещения *offset* файла (или другого объекта), определенного файловым описателем *fd*, в память, начиная с адреса *start*. Последний параметр (адрес) необязателен, и обычно бывает равен 0. Настоящее местоположение отраженных данных возвращается самой функцией mmap, и никогда не бывает равным 0. При удачном выполнении mmap возвращает указатель на область с отраженными данными. При ошибке возвращается значение MAP\_FAILED (-1), а переменная *errno* приобретает соответствующее значение. Флаги, использованные в работе:

PROT\_READ|PROT\_WRITE для записи и чтения

MAP\_SHARED|MAP\_ANONYMOUS Первый флаг нужен для того, чтобы разделить использование с другими процессами. Второй флаг обозначает следующее: отображение не резервируется ни в каком файле. Он необходим, поскольку мы не используем файл.

int munmap(void \**start*, size\_t *length*) - удаляет все отражения из заданной области памяти, после чего все ссылки на данную область будут вызывать ошибку "неправильное обращение к памяти" (invalid memory reference). Отражение удаляется автоматически при завершении процесса. С другой стороны, закрытие файла не приведет к снятию отражения. Start – адрес, по которому необходимо очистить память, length – размер памяти, которую необходимо очистить. При удачном выполнении munmap возвращаемое значение равно нулю. При ошибке возвращается -1, а переменная *errno* приобретает соответствующее значение. (Вероятнее всего, это будет EINVAL).

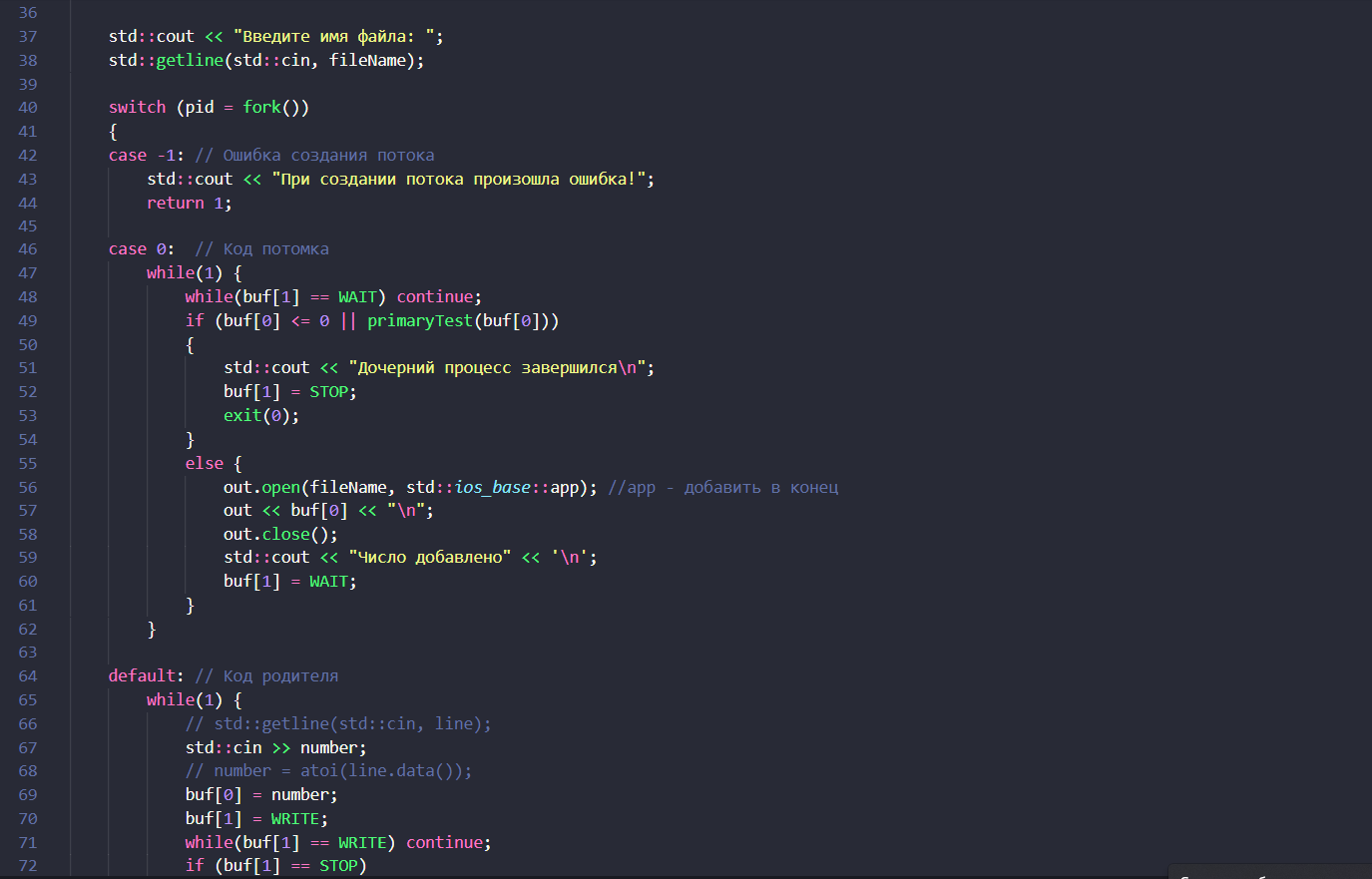
**Общий метод и алгоритм решения**

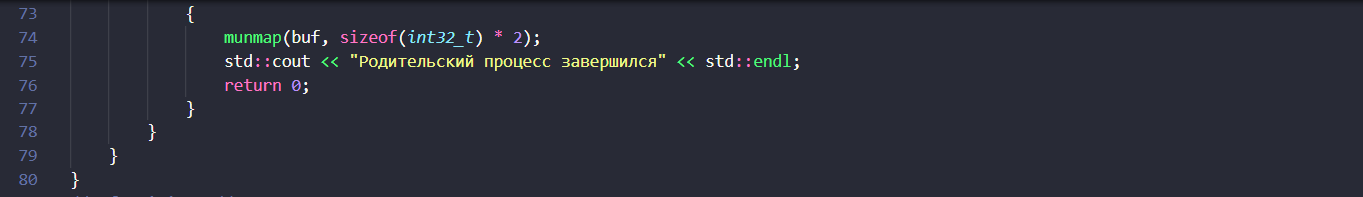
Программа получает на вход имя файла, потом число. Число и кодовое слово записываются родительским процессом в массив, созданный через mmap, после чего родительский процесс ждет дочерний, тот в свою очередь проверяет число. Если оно удовлетворяет условиям задачи (составное), то оно записывается в файл, дочерний процесс записывает в массив кодовое слово и родительский процесс продолжает свою работу, дочерний ожидает. Если же число не соответствует условиям задачи (простое), то дочерний процесс записывает кодовое слово STOP в массив и завершается, после чего родительский процесс принимает кодовое слово и завершается.

**Исходный код**

lab1.cpp







**Демонстрация работы программы**

Тест 1:



Тест 2:



Тест 3:



**Выводы**

Благодаря данной лабораторной работе я освоил принцип работы с файловыми системами и научился обеспечивать обмен данными между процессами посредством технологии «File mapping».