**Отчёт по лабораторной работе №4**

**по дисциплине “Операционные системы”**

Работу выполнил(а) студент(ка) группы М8О-206Б-20

Фурлетова Дарья Александровна, № по списку: 24

Контакты: e-mail: dafurletova@mail.ru

Работа сдана: 27 ноября 2021 г.

Преподаватель: Соколов А.А.

Итоговая оценка: \_\_\_\_\_.

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_.

1. **Тема.**

Процессы и потоки. Взаимодействие между потоками.

1. **Цель работы.**

Приобретение практических навыков в:

* освоении принципов работы с файловыми системами;
* обеспечении обмена данных между процессами посредством технологии «File mapping».

1. **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) должна создать для решения 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.

**Группа вариантов: 2.**

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия файла с таким именем на чтение. Стандартный поток ввода дочернего процесса переопределяется открытым файлом. Дочерний процесс читает команды из стандартного потока ввода.

Стандартный поток вывода дочернего процесса перенаправляется в pipe1. Родительский процесс читает из pipe1 и прочитанное выводит в свой стандартный поток вывода. Родительский и дочерний процесс должны быть представлены разными программами.

**Вариант № 7.**

В файле записаны команды вида: «число число число<endline>». Дочерний процесс считает их сумму и выводит результат в стандартный поток вывода. Числа имеют тип float. Количество чисел может быть произвольным.

1. **Описание программы.**

Т.к. нам необходимо работать с временным файлом, надо его создать при помощи mkstemp по шаблону template. Затем, при успешном создании временного файла установим ему длину в один float при помощи ftruncate, т.к. больше нам не нужно. Затем, как и во 2-ой лабораторной работе, для реализации данной задачи необходимо считать со стандартного ввода название файла, затем можно записать его в массив char-ов и после выдать этому файлу права на чтение.

Затем создаем дочерний процесс, направляем ему на стандартный ввод файл, потом запускаем child программу. Затем, снова как во 2-ой лабораторной работе, пока не встретится какой-то неправильный символ мы считываем поданный ему файл посимвольно, проверяя, что конкретно мы считали и делая соответствующие действия (например, если был считан ‘-’, надо его запомнить, чтобы потом учесть, что число было отрицательным; и т. д.). Если был считан какой-то неправильный символ, мы завершаемся с ошибкой, если же всё считалось корректно (т.е. не завершилось с ошибкой ещё в проверке считанного символа или позже), то мы открываем временный файл, созданный еще в главной программе на чтение и запись. При помощи mmap проецируем файл в оперативную память, затем записываем полученное значение суммы в этот файл и синхронизируем его содержание с самим файлом на диске.

Родительский процесс, увидев, что дочерний завершился успешно, проецирует временный файл в оперативную память, печатает оттуда результат суммирования и снимает отражение файла из памяти.

1. **Тестирование.**

| Входные данные | | Результат |
| --- | --- | --- |
| Название файла | Содержимое файла |
| test1 | -0.45 0.2 0.355 | 0,105 |
| test2 | 0.45 0.2 0.355 | 1,005 |
| test3 | -0.45 -0.2 -0.355 | -1,005 |
| test4 | 12 -3.45 0.2 | 8,75 |
| test5 | aafa | Wrong value in test\_file. |
| test6 |  | Test should end with <endline>. |
| test7 | ---  (такого файла нет) | File cannot be opened.: No such file or directory |

1. **Листинг программы.**

Продемонстрирован во время сдачи.

1. **Вывод.**

В ходе выполнения данной лабораторной работы, я освоила обмен данными между процессами посредством технологии «File mapping», научилась проецировать файлы в оперативную память, менять в них какие-то значение и записывать все изменения на файл на диске, также закрепила принципы перенаправления стандартных выводов и вводов программ.