**Отчёт по лабораторной работе №2**

**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.

1. **Цель работы.**

Приобретение практических навыков в:

* управлении процессами в ОС;
* обеспечении обмена данных между процессами посредством каналов.

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 каналы (pipe).

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.**

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 необходимо создать pipe, а после, при помощи fork запустить новый процесс. Дочерний процесс при помощи dup2 прикрепляет к стандартному вводу child текстовый файл, а к выводу - начало pipe, а затем execl запускает саму дочернюю программу.

В дочерней программе, 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 направляем через pipe полученный результат.

Родительский процесс, увидев, что дочерний завершился успешно, считывает из pipe значение суммы, которое посчитала дочерняя программа и выводит его пользователю.

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 познакомилась с созданием дочерних процессов и особенностями их выполнения (например, если после fork написать какой-нибудь вывод, то он выведется дважды, т.к. по нему пройдется и родительский и дочерний процесс), кроме того, я научилась создавать pipe и лучше стала понимать принципы перенаправления выводов и вводов программ и самого pipe.