



Отчет по лабораторной работе №3 по курсу

Операционные системы

Студент группы: **М8О-201Б-22**, **Парфенов Михаил Максимович**, № по списку: **21**, Контакты

mishaslsk@gmail.com Работа выполнена: « » _____ 202__ г.

Преподаватель: **Миронов Евгений**, Входной контроль знаний с

оценкой _____

Отчет сдан « » _____ 202__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Файлы, отображаемые в память
2. **Цель работы:** : Цель работы - приобретение практических навыков в управление процессами в ОС и обеспечение обмена данных между процессами посредством memory map
3. **Задание (вариант № 8):** В файле записаны команды вида: «число число число». Дочерний процесс производит деление первого числа команда, на последующие числа в команде, а результат выводит в стандартный поток вывода. Если происходит деление на 0, то тогда дочерний и родительский процесс завершают свою работу. Проверка деления на 0 должна осуществляться на стороне дочернего процесса. Числа имеют тип int. Количество чисел может быть произвольным
4. **Оборудование**
Оборудование ПЭВМ студента, если использовалось:
Процессор **AMD Ryzen 5 5500U** с ОП **8192 Мб**, ТТН **256 GB**. Мониторы **Dell Monitor, 1920 x 1080, 14 дюймов.**
5. **Программное обеспечение (лабораторное):**
Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства **GNU/Linux**, наименование **Arch Linux** версия **5.19.7** интерпретатор команд zsh версия 5.8.
Система программирования _версия _
Редактор текстов _ версия _
Утилиты операционной системы _
Прикладные системы и программы _
Местонахождение и имена файлов программ и данных на домашнем компьютере _
6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Логика работы аналогична логике из ЛР1: main.c делает fork и создает дочерний процесс с программой child.c, создается mmap для передачи данных между процессорами вида ключ - значение. Ключ отвечает за то, идет ли в маме запись или чтение, это было сделано чтобы предотвратить гонку в данном участке памяти. При делении на 0 child.c заносит в ключ значение 3. Если ввод окончен main.c заносит в ключ код 2 для сигнализации child.c завершения работы и началу записи в файл.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию]. *Пункты 1-7 отчета составляются строго до начала лабораторной работы.*
1. Изучить работу с `memory map`.
 2. Написать программу `main.c` как родительский процесс.
 3. Написать программу `child.c` как дочерний процесс.
 4. Скомпилировать и протестировать программу.

8. **Вывод:** Использование мапов оказывается более удобным по сравнению с пайпами. При работе с ними требуется создавать в два раза меньше файловых дескрипторов, поскольку они функционируют как массивы, а не как потоки данных. Единственная сложность при работе с мапами заключается в необходимости предотвращения гонок, но эту проблему можно решить с помощью сигналов, общих мьютексов или, как в моем случае, использования флага в начале `mmap`.