Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

Московский авиационный институт

(национальный исследовательский университет)

**ЛАБОРАТОРНАЯ РАБОТА №2**

по курсу

«Операционные системы»

Студент: Эссаулов А.А. (№31)

Группа: М80-207Б-18

Преподаватель: Миронов Евгений Сергеевич

Оценка:

Дата:

**Цель работы:**

Приобретение практических навыков в:

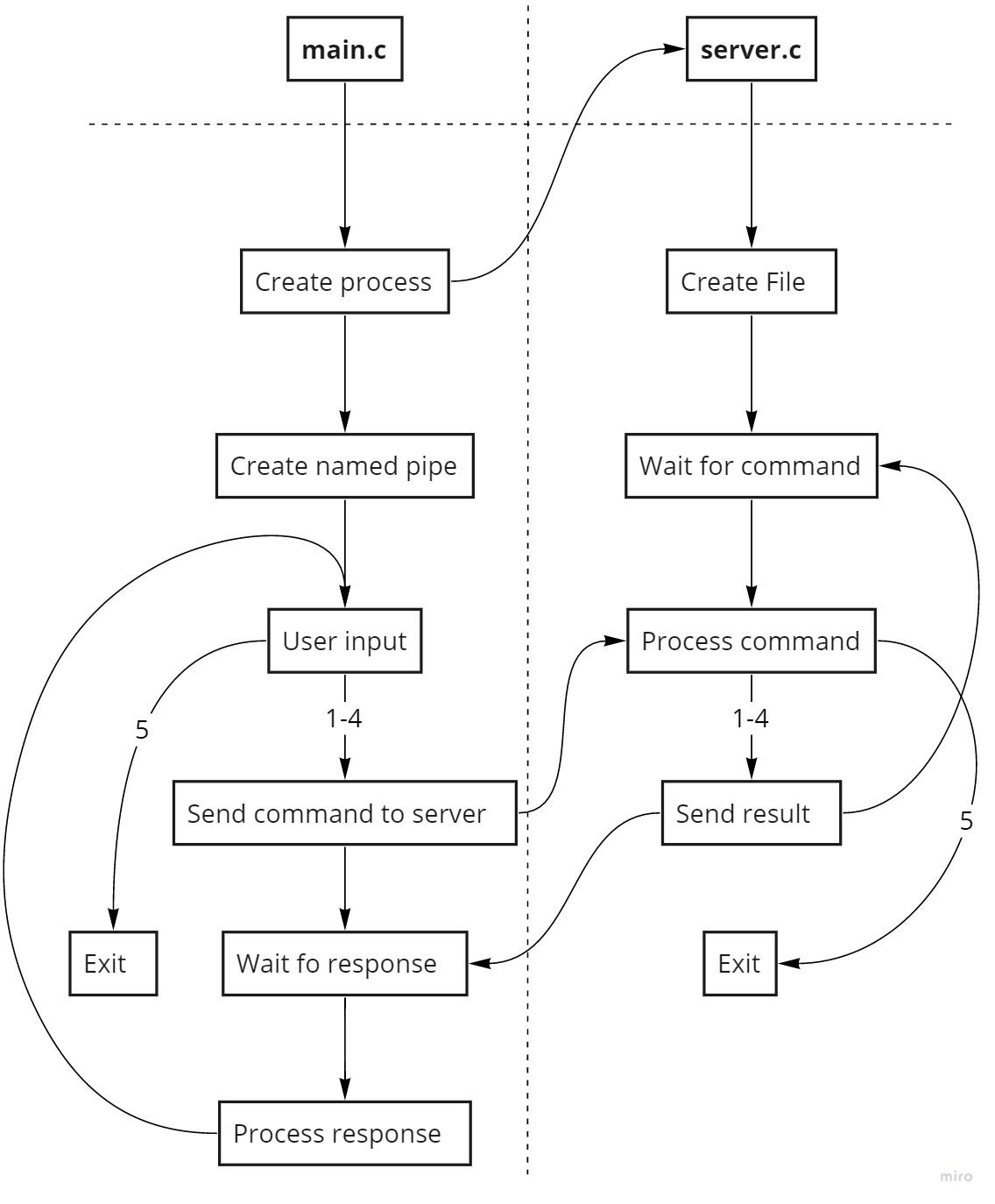
⦁ Управление процессами в ОС

⦁ Обеспечение обмена данных между процессами посредством каналов

**Задание (вариант № 19):**

Дочерний процесс представляет собой сервер по работе с бинарным деревом и принимает команды со стороны родительского процесса.

**Принцип работы:**



Программа работает по принципу клиент-серверного приложения, где клиентом является родительский процесс, а сервером дочерний процесс. Из родительского процесса происходит пользовательский ввод, который разбивается на команду и аргумент для отправки.

Формат команды - <код операции>.<аргумент>

Коды операций:

1) Создание корня дерева + аргумент

2) Добавление нода в дерево + аргумент

3) Удаление нода из дерева + аргумент

4) Вывод дерева

5) Выход

**Вывод:**

Изначально была предпринята попытка написания программы через анонимные пайпы, но наследование стандартного ввода и вывода работает слишком специфично. Кроме того, анонимный пайп всего лишь обертка на именованным пайпом, а значит использование именованного пайпа дает большей гибкости в работе.

Ловля ошибок дело крайне специфичное при работе с ОС. При работе с анонимными пайпами, когда стандартный вывод назначается на вход к родительскому процессу, так как отсутствует возможность сделать вывод в консоль со значениями переменных или номеров ошибок. А если ещё не было настроено клиент-серверное взаимодействие, то приходится использовать различные трюки для определения характера ошибок. Например, для определения заходит ли программа в определенный участок кода, можно поставить там выход из программы и тогда через диспетчер процессов, либо ProcMon можно увидеть, что программа завершилась. Либо, можно создавать файлы, а в качестве имен передавать значения переменных.

В общем работа с ОС дело крайне креативное и требующее понимания процессов происходящих после вызова каждой комманды.