Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №6,7,8 по курсу**

**«Операционные системы»**

Студент: Царев Михаил Вадимович

Группа: М8О-208Б-21

Вариант: 3

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**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/de1phin/OS/lab6

**Постановка задачи**

**Цель работы**

Целью является приобретение практических навыков в:

1. Управлении серверами сообщений (№6)
2. Применение отложенных вычислений (№7)
3. Интеграция программных систем друг с другом (№8)

**Задание**

Реализовать распределенную систему по асинхронной обработке запросов. В данной распределенной системе должно существовать 2 вида узлов: «управляющий» и «вычислительный». Необходимо объединить данные узлы в соответствии с той топологией, которая определена вариантом. Связь между узлами необходимо осуществить при помощи технологии очередей сообщений. Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом. При убийстве («kill -9») любого вычислительного узла система должна пытаться максимально сохранять свою работоспособность, а именно все дочерние узлы убитого узла могут стать недоступными, но родительские узлы должны сохранить свою работоспособность. Управляющий узел отвечает за ввод команд от пользователя и отправку этих команд на вычислительные узлы. Список основных поддерживаемых команд:

* **Создание нового вычислительного узла**

Формат команды: create id [parent]

id – целочисленный идентификатор нового вычислительного узла

parent – целочисленный идентификатор родительского узла. Если топологией не предусмотрено введение данного параметра, то его необходимо игнорировать (если его ввели)

Формат вывода:

«Ok: pid», где pid – идентификатор процесса для созданного вычислительного узла

«Error: Already exists» - вычислительный узел с таким идентификатором уже существует

«Error: Parent not found» - нет такого родительского узла с таким идентификатором

«Error: Parent is unavailable» - родительский узел существует, но по каким-то причинам с ним не удается связаться

«Error: [Custom error]» - любая другая обрабатываемая ошибка

* **Удаление существующего вычислительного узла**

Формат команды: remove id

id – целочисленный идентификатор удаляемого вычислительного узла.

Формат вывода:

«Ok» - успешное удаление

«Error: Not found» - вычислительный узел с таким идентификатором не найден

«Error: Node is unavailable» - по каким-то причинам не удается связаться с вычислительным узлом

«Error: [Custom error]» - любая другая обрабатываемая ошибка

* **Исполнение команды на вычислительном узле**

Формат команды: exec id [params]

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

Формат вывода:

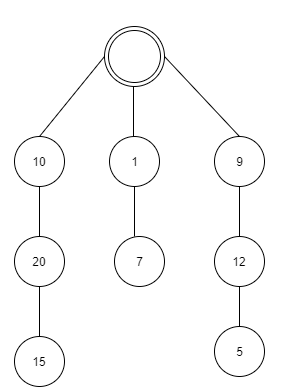
«Ok:id: [result]», где result – результат выполненной команды

«Error:id: Not found» - вычислительный узел с таким идентификатором не найден

«Error:id: Node is unavailable» - по каким-то причинам не удается связаться с вычислительным узлом

«Error:id: [Custom error]» - любая другая обрабатываемая ошибка

**Топология 1:**



Все вычислительные узлы находятся в списке. Есть только один управляющий узел. Чтобы

добавить новый вычислительный узел к управляющему, то необходимо выполнить команду:

create id 0.

**Набора команд 1** (подсчет суммы n чисел):

Формат команды: exec id n k1 … kn

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

n – количество складываемых чисел (от 1 до 10 8)

k1 … kn – складываемые числа

Пример:

> exec 10 3 1 2 3

Ok:10: 6

**Команда проверки 3:**

Формат команды: heartbit time

Каждый узел начинает сообщать раз в time миллисекунд о том, что он работоспособен. Если от

узла нет сигнала в течении 4\*time миллисекунд, то должна выводится пользователю строка:

«Heartbit: node id is unavailable now», где id – идентификатор недоступного вычислительного узла.

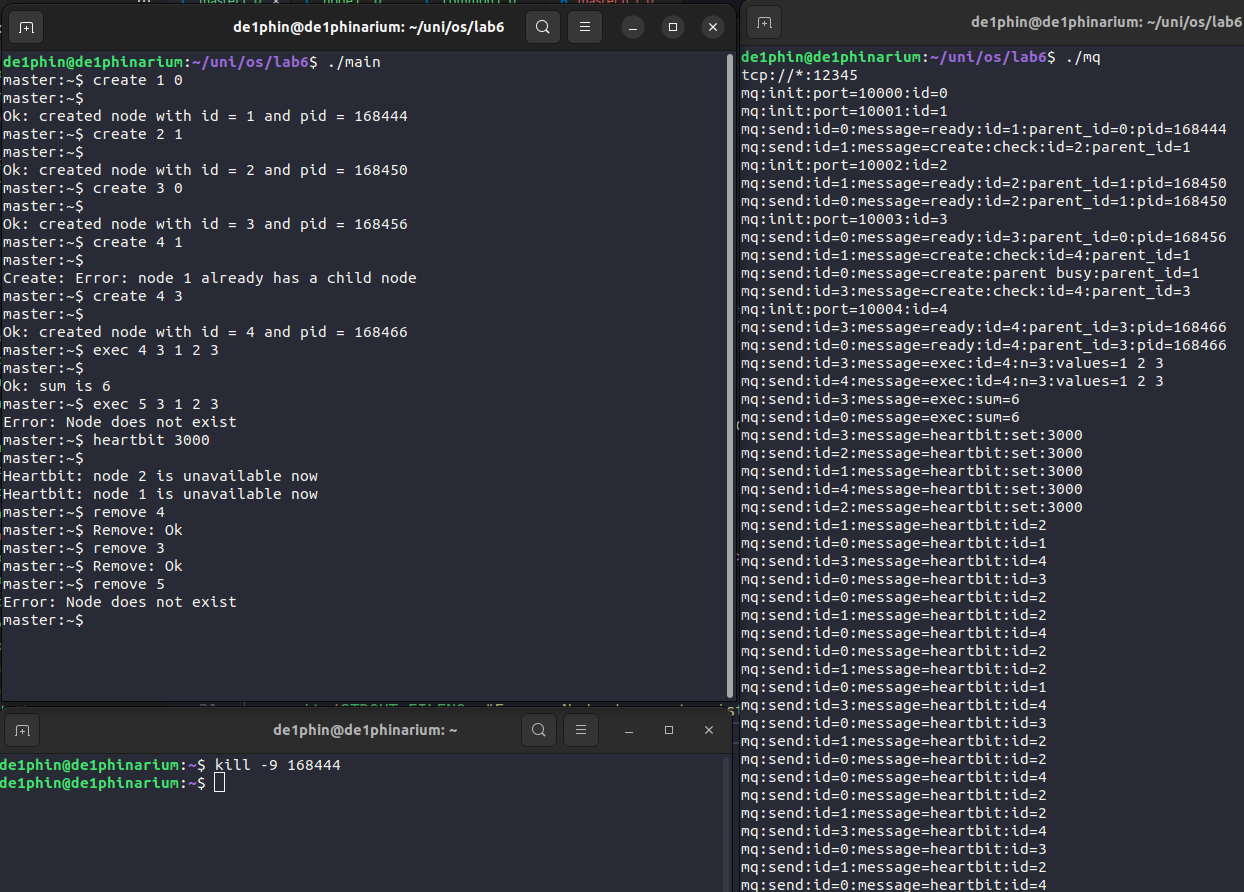
**Общие сведения о программе**

Программа состоит из master.c для управляющего узла, node.c для вычислительного узла и common.c для общего для них функционала. Общение происходит с помощью брокера и zmq. Брокер (broker/broker.c|broker.h) позволяет узлам легко общаться между собой лишь по id (но только в соответствии с топологией), без хлопот на переподключения zmq.

**Исходный код**

В репозитории.

**Демонстрация работы программы**



**Выводы**

Реализовал распределенную вычислительную систему с одним управляющим узлом, реализующую указанные функции, и осуществляющие коммуникацию в соответствии с топологией при помощи самописного брокера и библиотеки zmq