Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №6-8 по курсу**

**«Операционные системы»**

**Тема работы**

**“Очереди сообщений”**

Студент: Клитная Анастасия Викторовна

Группа: М8О-208Б-20

Вариант: 38

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**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/klitnaya/OS

**Постановка задачи**

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.  
  
Вариант 38. Команды:  
create id  
exec id subcommand(start |stop|time)

heartbeat time

**Общие сведения о программе**

Для выполнения данной лабораторной работы я предварительно реализовал 6 файлов с кодом:  
  
topology.h – реализация 1 Топологии(списка списков)

zmq\_functions.h - отдельный файл для функций zero-message queue, сделанный для удобства работы и во избежание загрязнения кода.  
  
control.cpp – файл работы с клиентом  
  
count.cpp - реализация программы клиента.

**Исходный код**

**Находится в репозийтории по ссылке**

[**https://github.com/klitnaya/OS/tree/main/os\_lab6-8/src**](https://github.com/klitnaya/OS/tree/main/os_lab6-8/src)

**Результат работы**root@Owl:/mnt/c/Users/Настя# ./control

create 89 -1

OK: 4656

create 90 -1

OK: 4659

create 100 90

OK: 4662

create 200 89

OK: 4665

heartbeat 1000

89:Ok; 200:Ok;

90:Ok; 100:Ok;

excec 200 time

Incorrect comand >excec<!

Incorrect comand >200<!

Incorrect comand >time<!

exec 200 time

OK: 200: 0

exec 200 start

OK: 200

exec 100 start

OK: 100

exec 200 stop

OK: 200

exec 100 stop

OK: 100

exec 100 time

OK: 100: 19626

exec 200 time

OK: 200: 13454

remove 100

OK

exec 100 time

Error: incorrect node id!

heartbeat 1000

89:Ok; 200:Ok;

90:Ok;

exit

root@Owl:/mnt/c/Users/Настя#

**Выводы**Данная лабораторная работа была очень и очень непростой. Во время ее выполнения я полностью осозналa концепцию очередей сообщений на основе zero message queue. На мой взгляд, это достойное завершение курса “Операционных систем”.