**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»**

**6. УПРАВЛЕНИЕ СЕРВЕРАМИ СООБЩЕНИЙ.**

**7. ПРИМЕНЕНИЕ ОТЛОЖЕННЫХ ВЫЧИСЛЕНИЙ.**

**8. ИНТЕГРАЦИЯ ПРОГРАММНЫХ СИСТЕМ ДРУГ С ДРУГОМ.**

**Студент: Баранников Степан Алексеевич**

**Группа: М80–201Б-18**

**Вариант: 16**

**Преподаватель: Соколов Андрей Алексеевич**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_**

**Подпись: \_\_\_\_\_\_\_\_\_\_\_**

**Москва, 2019**

**1.** **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.

**Топология:** дерево общего вида (реализована цепочка)

**Тип команд:** Cумма n чисел

**Тип проверки доступности узлов:** *heartbeat id*

**2.Описание**

**Controller.cpp** содержит исходный код для управляющего узла. После его запуска и создания портов для общения с вычислительными узлами, программа входит в цикл, который продолжает работать до получения команды *“q”*. Он также принимает следующие команды:

· *“create [id]”*

Создает вычислительный узел и привязывает его к последнему доступному узлу.

· *“ping [id]”*

Отправляет вычислительному узлу запрос, после чего ожидает заданное количество времени. Если ответ пришел, то возвращает успех, иначе предупреждает об истечении времени.

· *“exec [id] [amount] [a b c d e ...]"*

Отправляет вычислительному узлу задание на подсчет суммы чисел [a, b, c, d...], равных количеству amount.

“*heartbeat [id] [time] [amount]*”

Пингует [amount] раз по [time] секунд.

**executor.cpp** – исходный код для вычислительного узла. Сразу после его создания он отправляет родителю сообщение об успехе и переходит в режим ожидания, когда один из входящих сокетов не получит сообщение, после чего узел просматривает адресата и принимает решение – если id адресата и собственный не совпадают, то отправить ниже по цепочке, иначе выполнить действие, описанное в сообщении.

**4.** **Вывод**

С помощью библиотеки ZeroMQ можно добиться реализации сервера сообщений, что дает нам множество возможностей для реализации большинства других многопользовательских проектов, ведь в ZeroMQ есть различные группы сокетов для общения узлов, подходящие для тех или иных задач.