МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота №1

з дисципліни «Системи моніторингу стану мережі об'єктів у реальному часі»

**Виконав:**

студент 5-го курсу

ТЕФ, гр. ТР-02мп

Круглий Д.В.

**Перевірив:**

проф. Сліпченко В. Г.

**Київ-2021**

**Тема:** модель клієнт-сервер.

**Мета:** побудова моделюючого комплексу для моделі клієнт-сервер.

**Постановка задачі:** потрібно запрограмувати завдання, написавши монітор "Сервер", який реалізує обслуговування клієнтів в порядку черговості їх запитів.

Для виконання даної лабораторної роботи мною була вибрана мова програмування JavaScript в середовищі виконання Node.js.

**Представлення коду програми**

**Client.js**

module.exports = class Client {

    constructor(server, id) {

        this.server = server;

        this.id = id;

        this.init();

    }

    SERVER\_REQUEST\_INTERVAL = 15000;

    init() {

        this.startTaskProcessing();

    }

    startTaskProcessing() {

        setInterval(() => {

            setTimeout(() => this.server.clientRequest(this.id), Math.random() \* 10000)

        }, this.SERVER\_REQUEST\_INTERVAL);

    }

}

**Server.js**

module.exports = class Server {

    static BUSY\_STATE = "BUSY"

    static FREE\_STATE = "FREE"

    CHECK\_INTERVAL = 100;

    clients = {};

    clientsQueue = [];

    state = {

        processingClientId: null

    }

    init() {

        setInterval(() => this.processClient(), this.CHECK\_INTERVAL);

    }

    addClient(c) { this.clients[c.id] = c; }

    isFree() { return !!this.state.isProcessing }

    addToQueue(c) { this.clientsQueue.push(c) }

    async processClient() {

        if ( this.state.processingClientId || !this.clientsQueue.length) return;

        const clientToProcess = this.clientsQueue.shift();

        this.state.processingClientId = clientToProcess.id;

        await new Promise(( res, rej ) => setTimeout(res, Math.random() \* 10000));

        this.state.processingClientId = null;

        console.log(`processed task from client ${clientToProcess.id}`);

        console.log(this.clientsQueue.length ?  `tasks in queue - ${this.clientsQueue.length}` : 'all tasks completed')

    }

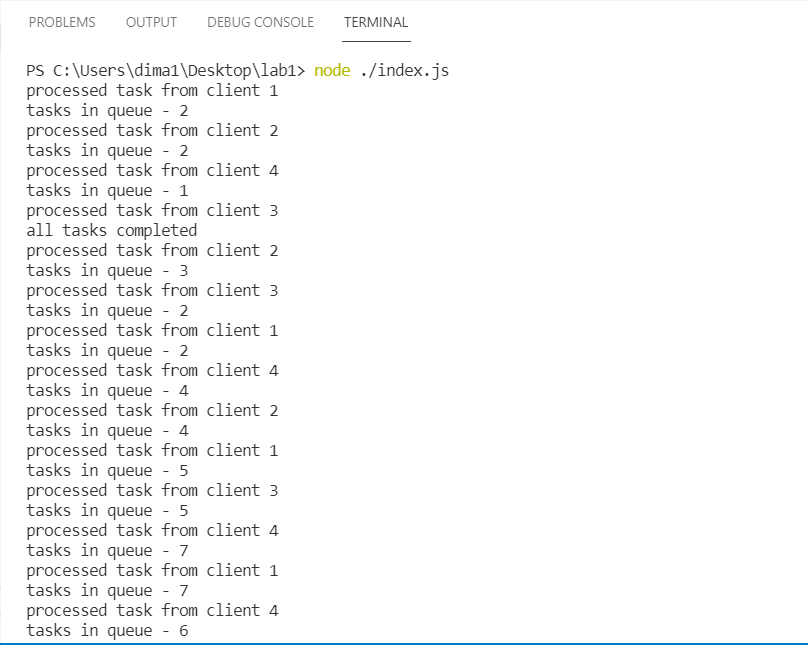
    clientRequest(id) {

        this.addToQueue(this.clients[id]);

    }

}

**Приклад роботи програми**



**Висновок:** виконавши дану лабораторну роботу, я побудував моделюючий комплекс для моделі клієнт-сервер. Також мною була вирішена задача з урахування декількох обслуговуючих процесів.