Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

ЗВІТ

за результатами виконаної роботи

з розрахунково-графічної роботи

дисципліни «Системний аналіз»

на тему «Системи масового облуговування»

|  |  |
| --- | --- |
| Виконав:  Студент групи ІТ-52  Кіндрат А. А.  20.12.2018  Захищено з балом \_\_\_\_\_\_\_\_\_\_\_ | Перевірив:  Доцент кафедри ТК  Чумаченко О. І. |

Київ 2018

ВСТУП 4

ТЕОРЕТИЧЕСКИЕ ВЕДОМОСТИ 6

МНОГОКАНАЛЬНАЯ СМО С ОГРАНИЧЕННОЙ ОЧЕРЕДЬЮ 7

ПРОГРАММА 8

КОД ПРОГРАММЫ 10

ОПИСАНИЕ СИСТЕМИ 15

ВЫВОД 16

ВСТУП

Системы массового обслуживания — это такие системы, в кото­рые в случайные моменты времени поступают заявки на обслужи­вание, при этом поступившие заявки обслуживаются с помощью имеющихся в распоряжении системы каналов обслуживания.

С позиции моделирования процесса массового обслуживания ситуации, когда образуются очереди заявок (требований) на обслу­живание, возникают следующим образом. Поступив в обслуживаю­щую систему, требование присоединяется к очереди других (ранее поступивших) требований. Канал обслуживания выбирает требова­ние из находящихся в очереди, с тем, чтобы приступить к его об­служиванию. После завершения процедуры обслуживания очеред­ного требования канал обслуживания приступает обслуживанию следующего требования, если таковое имеется в блоке ожидания.

Цикл функционирования системы массового обслуживания подоб­ного рода повторяется многократно в течение всего периода рабо­ты обслуживающей системы. При этом предполагается, что пере­ход системы на обслуживание очередного требования после завершения обслуживания предыдущего требования происходит мгновенно, в случайные моменты времени.

Примерами систем массового обслуживания могут служить:

•  посты технического обслуживания автомобилей;

•  посты ремонта автомобилей;

•  станции технического обслуживания автомобилей;

•  аудиторские фирмы;

* телефонные станции и т. д.

**Цель даной работы:** разработать пз которое будет моделировать систему массового обслуживания. СМО многоканальную с ограничением на длину очереди.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Вхідний потік та його характеристики | Параметри СМО | Характеристика черги | Точність та достовірність |
| Рівномірне, λ = 7 | µ = 2 , n = 5 | m = 7 | ε = 0,03 , α = 0,97 |

Cредой разработки было выбрано язык JavaScript. IDE - sublime text + google chrome for testing.

ТЕОРЕТИЧЕСКИЕ ВЕДОМОСТИ

**Система массового обслуживания** (СМО) — система, которая производит обслуживание поступающих в неё требований. Обслуживание требований в СМО осуществляется обслуживающими приборами. Классическая СМО содержит от одного до бесконечного числа приборов. В зависимости от наличия возможности ожидания поступающими требованиями начала обслуживания СМО подразделяются на:

* системы с потерями, в которых требования, не нашедшие в момент поступления ни одного свободного прибора, теряются;
* системы с ожиданием, в которых имеется накопитель бесконечной ёмкости для буферизации поступивших требований, при этом ожидающие требования образуют очередь;
* системы с накопителем конечной ёмкости (ожиданием и ограничениями), в которых длина очереди не может превышать ёмкости накопителя; при этом требование, поступающее в переполненную СМО (отсутствуют свободные места для ожидания), теряются.

Выбор требования из очереди на обслуживание производится с помощью так называемой дисциплины обслуживания. Их примерами являются FCFS/[FIFO](https://ru.wikipedia.org/wiki/FIFO) (пришедший первым обслуживается первым), LCFS/[LIFO](https://ru.wikipedia.org/wiki/LIFO) (пришедший последним обслуживается первым), random (случайный выбор). В системах с ожиданием накопитель в общем случае может иметь сложную структуру.

**Основные понятия**

**Требование (заявка) — запрос на обслуживание.**

* **Входящий поток требований** — совокупность требований, поступающих в СМО.
* **Время обслуживания** — период времени, в течение которого обслуживается требование.
* **Математическая модель СМО** — это совокупность математических выражений, описывающих входящий поток требований, процесс обслуживания и их взаимосвязь.

МНОГОКАНАЛЬНАЯ СМО С ОГРАНИЧЕННОЙ ОЧЕРЕДЬЮ

Многоканальная СМО с ограниченной длиной очереди.

Система может находиться в одном из состояний S0, S1, S2,…, Sk,…, Sn,…, — нумеруемых по числу заявок, находящихся в СМО: S0 — в системе нет заявок (все каналы свободны); S1 — занят один канал, остальные свободны; S2 — заняты два канала, остальные свободны;..., Sk — занято k каналов, остальные свободны;..., Sn — заняты все n каналов (очереди нет); Sn+1 — заняты все n каналов, в очереди одна заявка;..., Sn+r — заняты все n каналов, r заявок стоит в очереди.



ПРОГРАММА

Мною было создано программное обеспечение, а в частности веб-приложение для построения графов и решения СМО.

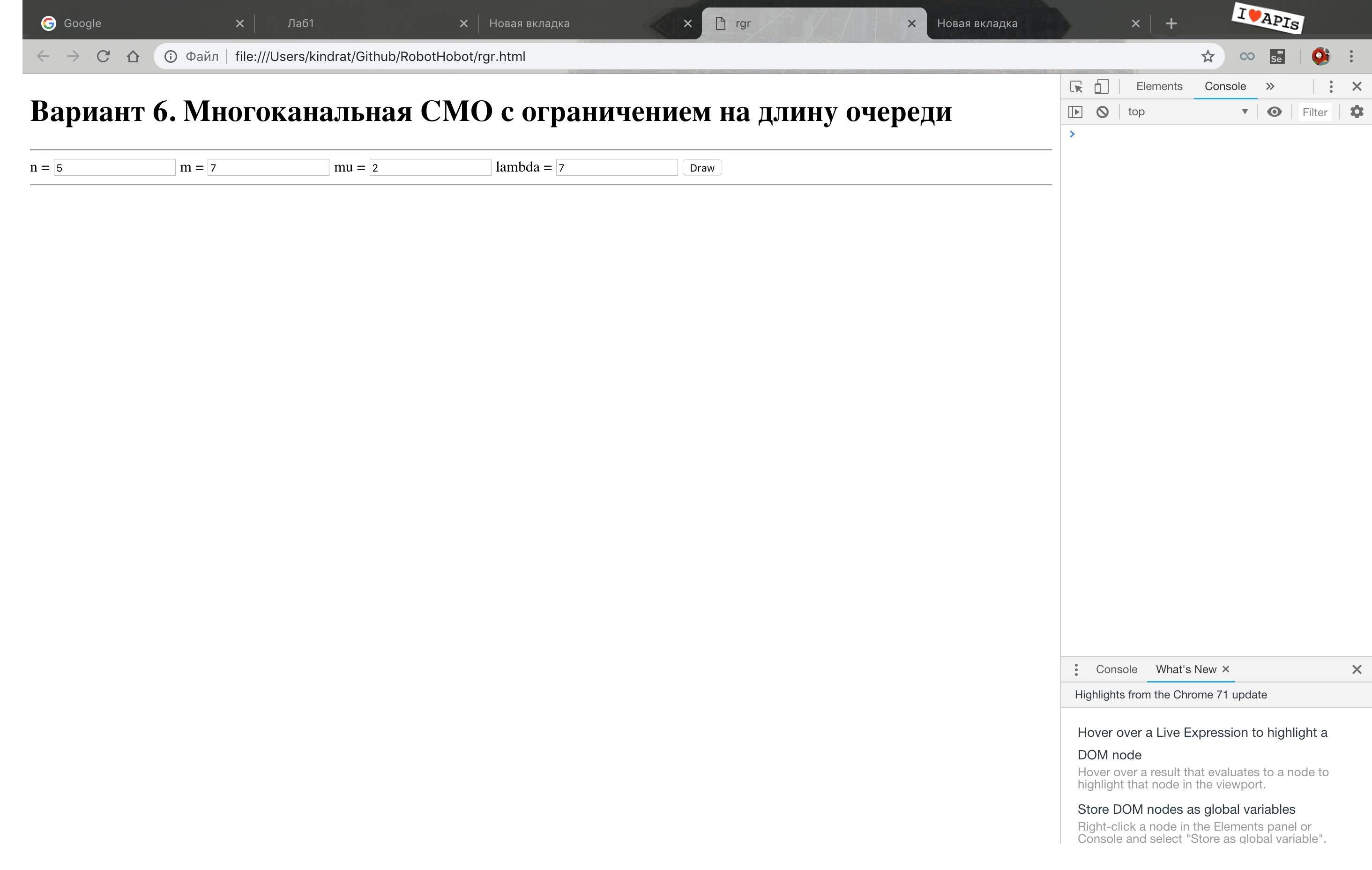
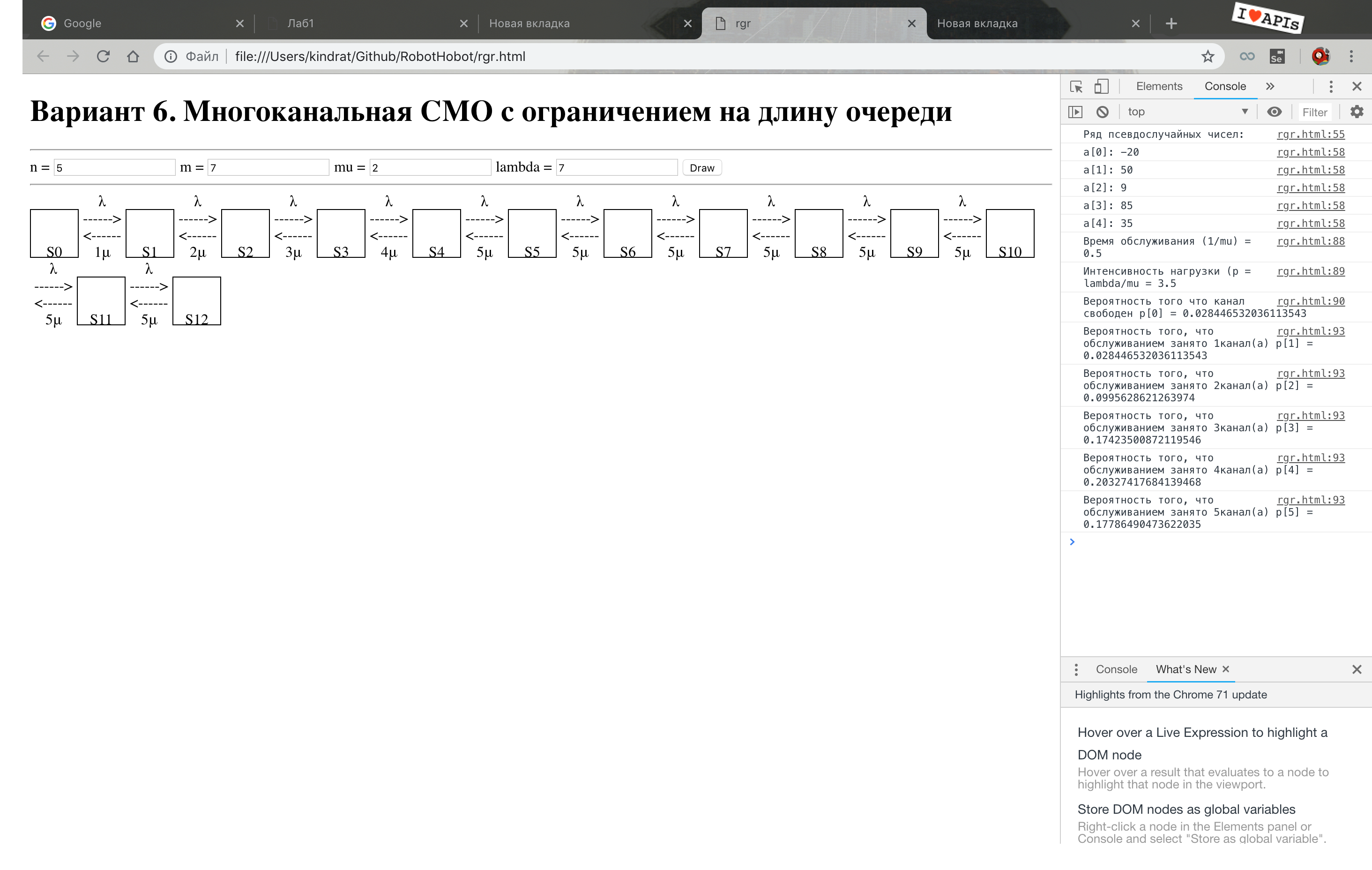
Программное обеспечение в управлении и понимании простым пользователем максимально юзерфрендли. Рисунок 1 демонстрирует простоту программного обеспечения

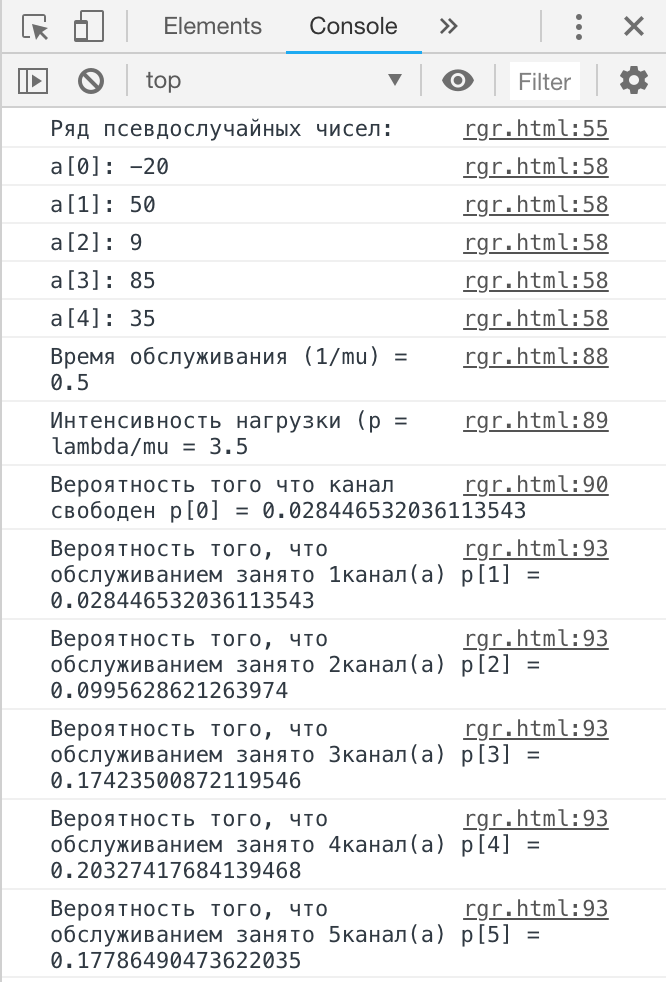
Рисунок 1. ПО многоканальной СМО с ограниченной очередью

Всего есть 4 поля для ввода:

* n - количество каналов
* m - очередь
* Mu - обслуживание одним каналом
* Lambda - интенсивность входного потока (надходження)
* Кнопка draw - построение графа и вычисления которые происходят в СМО

После нажатия кнопки draw - строится граф и производятся вычисления как показано на Рисунке 2

Рисунук 2. Программа в состоянии «Произведены вычисления»

В боковом меню указаны все расчеты и сгенерирован ряд псевдорандомных чисел как показано на Рисунке 3. «Вычисления»

КОД ПРОГРАММЫ

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>

<title>rgr</title>

<style>

.container{

height:100px;

display:inline-block;

}

.block{

width:50px;

height:50px;

background:white;

text-align:center;

display:inline-block;

border:1px solid black;

}

.blockStrelka{

width:50px;

height:50px;

background:white;

text-align:center;

display:inline-block;

}

</style>

<div>

<script>

$(document).ready(function(){

$('#Draw').click(function(){

let n = $("#n").val();

let m = $("#m").val();

let mu = $("#mu").val();

let lambda = $("#lambda").val();

let sum = parseInt(n)+parseInt(m)+1;

let sumForRow = 0;

let field = document.getElementById("TheList");

let po = lambda/mu;

function fact(x) {

if(x==0) {

return 1;

}

return x \* fact(x-1);

}

function getRandomInt(min, max)

{

return Math.floor(Math.random() \* (max - min + 1)) + min;

}

console.log('Ряд псевдослучайных чисел: ');

for(let i = 0;i<parseInt(n);i++){

randomNumber = getRandomInt(-100, 100);

console.log('a[' + i + ']: ' + randomNumber);

}

$('#blockStrelka').append('<div class="block"><br><br>S0<br></div>');

for(let i=0;i<parseInt(n);i++){

$('#blockStrelka').append('<div class="blockStrelka">&lambda;<br>------><br><------<br>'+(i+1)+'&mu;</div><div class="block"><br><br>S'+(i+1)+'<br></div>');

}

for(let i=0;i<parseInt(m)-1;i++){

$('#blockStrelka').append('<div class="blockStrelka">&lambda;<br>------><br><------<br>'+n+'&mu;</div><div class="block"><br><br>S'+(i+1+parseInt(n))+'<br></div>');

}

$('#blockStrelka').append('<div class="blockStrelka">&lambda;<br>------><br><------<br>'+n+'&mu;</div><div class="block"><br><br>S'+(parseInt(n)+parseInt(m))+'<br></div>');

let sum1 = 0;

for(let i=0;i<parseInt(n);i++){

//sumForRow = (1/((Math.pow(po,i)/fact(i)) + (Math.pow(po,(n+1))/(fact(n)\*(n-po)))\*(1-Math.pow((po/n),m))));

n1 = Math.pow(po,i);

n2 = n1/fact(i);

sum1+=n2;

}

n3 = Math.pow(po,(parseInt(n)+1));

n4 = fact(parseInt(n))\*(parseInt(n)-po);

n5 = n3/n4;

n6 = 1 - Math.pow((po/parseInt(n)),parseInt(m));

n7 = 1/(sum1 +n5+ n6);

console.log('Время обслуживания (1/mu) = ' + (1/mu));

console.log('Интенсивность нагрузки (p = lambda/mu = ' + po);

console.log('Вероятность того что канал свободен p[0] = ' + n7);

for(i=0;i<parseInt(n);i++){

p = Math.pow(po,i)/fact(i) \* n7;

console.log("Вероятность того, что обслуживанием занято " + (i+1) + "канал(а) p[" + (i+1) + "] = " + p);

}

/\*var S=0, e;

for ( var n=1; n++) {

e=

//if (e<0.001) { break;}

S+=e;

document.write("sum=",S); \*/

});

});

</script>

<div>

</head>

<body>

<div>

<h1>Вариант 6. Многоканальная СМО с ограничением на длину очереди</h1>

<hr>

n = <input type="number" value=5 id="n">

m = <input type="number" value=7 id="m">

mu = <input type="number" value=2 id="mu">

lambda = <input type="number" value=7 id="lambda">

<input type="button" id="Draw" value="Draw">

<hr>

</div>

<div id="blockStrelka"></div>

</body>

</html>

ОПИСАНИЕ СИСТЕМИ

Система работает в правильном порядке. Все вычисления были проверены вручную.

Система может находиться в одном из состояний S0, S1, S2,…, Sk,…, Sn,…, — нумеруемых по числу заявок, находящихся в СМО: S0 — в системе нет заявок (все каналы свободны); S1 — занят один канал, остальные свободны; S2 — заняты два канала, остальные свободны;..., Sk — занято k каналов, остальные свободны;..., Sn — заняты все n каналов (очереди нет); Sn+1 — заняты все n каналов, в очереди одна заявка;..., Sn+r — заняты все n каналов, r заявок стоит в очереди.



ВЫВОД

Итого, благодаря данной работе я улучшил свои знания в понимании СМО и произведении вычислений и работой с многоканальными системами массового обслуживания с ограничением на длину очереди.

Мною было разработано программное обеспечение которое и показывает мои практические навыки!