МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Исследование операций»

**Лабораторная работа №1.**

**Система массового обслуживания**

**одноканальная модель**

Выполнил:

студент группы ИВТАПбд-21

Кондратьев П. С.

Проверил:

Фролов В. А.

Ульяновск, 2018

**Введение**

***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 к обслуживанию следующего требования, если такое 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:

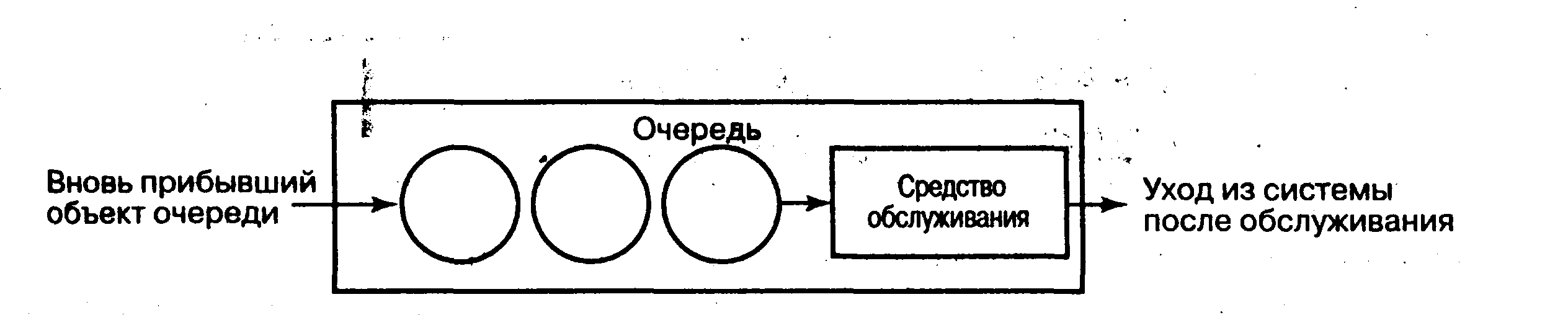
* посты технического обслуживания автомобилей;
* посты ремонта автомобилей;
* персональные компьютеры, обслуживающие поступающие за­явки или требования на решение тех или иных задач;
* станции технического обслуживания автомобилей;
* аудиторские фирмы;
* отделы налоговых инспекций, занимающиеся приемкой и проверкой текущей отчетности предприятий;
* телефонные станции и т. д.

Основными компонентами системы массового обслуживания любого вида являются:

* входной поток поступающих требований или заявок на обслу­живание;
* дисциплина очереди;
* механизм обслуживания.

**Простейшая одноканальная модель**

Такой моделью с вероятностными входным потоком и процедурой обслуживания является модель, характеризуемая показательным распределением как длительностей интервалов между поступлениями требований, так и длительностей обслуживания.



**Задача**

В вычислительном центре работает 5 персональных компьютеров (ПК). Простейший поток задач, поступающих на ВЦ, имеет интенсивность λ=10 задач в час. Среднее время решения задачи равно 12 мин. Заявка получает отказ, если все ПК заняты. Найдите вероятностные характеристики системы обслуживания (ВЦ).

**Решение**

Определим исходные данные. Интенсивность входящего потока λ=10 задач в час=10/60=1/6 в минуту. Среднее время обслуживания (решения задачи) t – 12 мин, значит интенсивность потока обслуживания. Число каналов n=5. Система имеет следующие состояния: (1) S0 – компьютеры свободны; (2) S1 – 1 компьютер занят, остальные свободны; (3) S2 – 2 компьютера заняты, остальные свободны; (4) S3 – 3 компьютера заняты, остальные свободны; (5) S4 – 4 компьютера заняты, остальной свободен; (6) S5 – все компьютеры заняты. Построим граф состояний (Рис. 1).

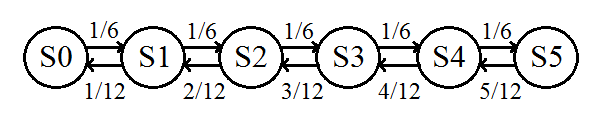


Рисунок 1. Граф состояний системы

Система переходит из любого левого состояния в следующее правое с интенсивностью 1/6. Интенсивность перехода из правого в левое состояние меняется в зависимости от состояния. Например, из состояния S2 (2 компьютера заняты) в S1 (один компьютер занят) система может перейти тогда, когда закончит работу либо второй, либо первый компьютер, то есть суммарная интенсивность их потоков обслуживания будет 2µ = 2/12.

Определим приведенную интенсивность входящего потока:

Предельные вероятности состояний найдем по формулам Эрланга:

Вероятность отказа в обслуживании заявки:

Относительная пропускная способность системы (ВЦ):

Абсолютная пропускная способность системы (ВЦ):

Среднее число занятых каналов:

**Интерпретация результатов**

Из вычисленных показателей видно, что в установившемся режиме работы системы в среднем будут заняты 2 канала (ПК) из 5, а остальные 3 будут простаивать. Вероятность отказа в обслуживании в среднем 3-4%, а значит, работу системы можно считать удовлетворительной.

**Исходный код**

let queue = [],

servers = [],

currentTime = 0.0,

prevTime,

// исходные данные

MAX\_QUEUE\_LENGTH = 5,

SERVERS\_NUMBER = 2,

LAMBDA = 1.0 / 5,

MU = 1.0 / 3,

SIMULATION\_DURATION = 100 \* 60,

// статистика

queueLengthDurations,

activeServerDurations,

emptyServerDuration = 0.0,

queueWaitingTime = 0.0,

eventCount = 0;

let init = function () {

// статистика

queueLengthDurations = new Array(MAX\_QUEUE\_LENGTH + 1);

for (let i = 0; i < queueLengthDurations.length; i++)

queueLengthDurations[i] = 0.0;

activeServerDurations = new Array(SERVERS\_NUMBER + 1);

for (let i = 0; i < activeServerDurations.length; i++)

activeServerDurations[i] = 0.0;

};

let processEvent = function (currentEvent) {

// статистика

queueLengthDurations[queue.length] += currentTime - prevTime;

activeServerDurations[servers.length] += currentTime - prevTime;

if (servers.length == 0) {

emptyServerDuration += currentTime - prevTime;

}

eventCount++;

// обновление состояния устройств обслуживания

servers = servers.filter(function (event) {

return event.end > currentTime;

});

while (servers.length < SERVERS\_NUMBER && queue.length > 0) {

let event = queue.shift();

event.end = currentTime + event.duration;

servers.push(event);

// статистика

queueWaitingTime += currentTime - event.begin;

}

// есть свободные устройства

if (servers.length < SERVERS\_NUMBER) {

currentEvent.end = currentTime + currentEvent.duration;

servers.push(currentEvent);

// свободных устройств нет, но есть место в очереди

} else if (queue.length < MAX\_QUEUE\_LENGTH) {

queue.push(currentEvent);

// нет мест в очереди

} else {

}

}

let statistics = function () {

let meanQueueLength = 0.0;

for (let i = 0; i < queueLengthDurations.length; i++) {

meanQueueLength += queueLengthDurations[i] \* i;

meanQueueLength /= SIMULATION\_DURATION;

}

console.log(`Средняя длина очереди: ${meanQueueLength}`);

let meanServerUsage = 0.0;

for (let i = 1; i < activeServerDurations.length; i++) {

meanServerUsage += activeServerDurations[i] \* i;

}

meanServerUsage /= SIMULATION\_DURATION;

console.log(`Cреднее число занятых каналов обслуживания: ${meanServerUsage}`);

console.log(`Вероятность отсутствия требований в системе: ${emptyServerDuration / SIMULATION\_DURATION}`);

console.log(`Среднее время ожидания обслуживания : ${queueWaitingTime / eventCount}`);

console.log(`Общее число событий : ${eventCount}`);

};

let exponentialDistribution = function (lambda) {

return -(Math.log(Math.random()) / lambda);

}

let main = function () {

init();

// генерация событий

while (currentTime < SIMULATION\_DURATION) {

let delay = exponentialDistribution(LAMBDA),

currentEvent = {};

prevTime = currentTime;

currentTime = currentTime + delay;

currentEvent.begin = currentTime;

currentEvent.duration = exponentialDistribution(MU);

processEvent(currentEvent);

}

statistics();

};

main();