

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Э. БАУМАНА

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»



---

# Моделирование

---

ЛАБОРАТОРНАЯ РАБОТА №4  
МОДЕЛИРОВАНИЕ ПРОСТЕЙШЕЙ СИСТЕМЫ

Студент: Петухов И.С.

Группа: ИУ7-71

Преподаватель: Рудаков И.В.

Москва, 2016

## Содержание

|     |  |   |
|-----|--|---|
| 1   | Аналитический раздел . . . . .           | 3 |
| 2   | Технологический раздел . . . . .         | 4 |
| 2.1 | Язык программирования . . . . .          | 4 |
| 2.2 | Примеры кода . . . . .                   | 4 |
| 2.3 | Взаимодействие с пользователем . . . . . | 6 |

## 1 Аналитический раздел

**Цель** данной работы - смоделировать систему, состоящую из генератора, источника информации, блока памяти и обслуживающего аппарата.

Закон генерации заявок - равномерный (параметры настраиваются и варьируются).

В ОА закон распределение Пуассона.

Определить оптимальный размер буферной памяти, т.е. ту длину, при которой ни одно сообщение необработанным не исчезает (т.е. нет отказа)

Должна быть возможность возвращения заявки в очередь после ее обработки.

Управляющую программу имитационной модели реализовать двумя подходами:

- событийный
- $\Delta t$  (ориентирован на действие)

## 2 Технологический раздел

### 2.1 Язык программирования

В качестве языка программирования выбран язык высокого уровня JavaScript.

### 2.2 Примеры кода

Листинг 2.1 — Событийный принцип: поиск минимального размера буферной памяти

```
1 function findMinLengthQueue(a, b, lambda, finishTime, releaseOrder) {
2   let si = new SourceOfInformation(a, b);
3   let su = new ServiceUnitByPoisson(releaseOrder, lambda);
4
5   let bmLength = 0;
6   let flagRun = true;
7
8   while (flagRun) {
9     ++bmLength;
10
11     try {
12       runSimulation(si, su, bmLength, finishTime);
13     } catch (err) {
14       flagRun = !flagRun;
15     } finally {
16       flagRun = !flagRun;
17     }
18   }
19
20   return bmLength;
21 }
22
23 function runSimulation(si, su, bmLength, finishTime) {
24
25   let bm = new Memory(bmLength);
26
27   let fel = [];
28
29   let blocks = [si, su];
30
31   initBlock(fel, blocks, finishTime);
32
33
34   while (true) {
35     let iMin = getNumberMin(fel);
36
```

```

37     if (iMin == blocks.length) {
38         break;
39     }
40
41     blocks[iMin].run(bm);
42     fel[iMin] += blocks[iMin].getNextTime()
43 }
44 }

```

Листинг 2.2 — Принцип dt: поиск минимального размера буферной памяти

```

1 function findMinLengthQueue(dt, a, b, lambda, finishTime, releaseOrder) {
2     let bmLength = 0;
3     let flagRun = true;
4
5     while (flagRun) {
6         ++bmLength;
7
8         try {
9             let si = new SourceOfInformation(a, b);
10            let su = new ServiceUnitByPoisson(releaseOrder, lambda);
11
12            let bm = new Memory(bmLength);
13
14            let nowTime = 0;
15            let blocks = [si, su];
16
17            while (nowTime < finishTime) {
18                blocks.forEach(block => {
19                    block.run(nowTime, bm);
20                })
21
22                nowTime += dt;
23            }
24        } catch (err) {
25            flagRun = !flagRun;
26        } finally {
27            flagRun = !flagRun;
28        }
29    }
30
31    return bmLength;
32 }

```

## 2.3 Взаимодействие с пользователем

Взаимодействие с пользователем осуществляется через html страницы, открытые в браузере.

---

**Управляющая программа имитационной модели (принцип dt)**

---

**Параметры имитационной модели**

время работы:

dt:

☒ Выпускать обработанные заявки

---

**Источник информации**

параметры равномерного распределения:

a:

b:

---

**Обслуживающий аппарат**

параметры распределения Пуассона:

lambda:

Минимальная длина буферной памяти:

---

найти минимальную длину буферной памяти

Рисунок 2.1 — Принцип  $\Delta t$ : поиск минимального размера буферной памяти

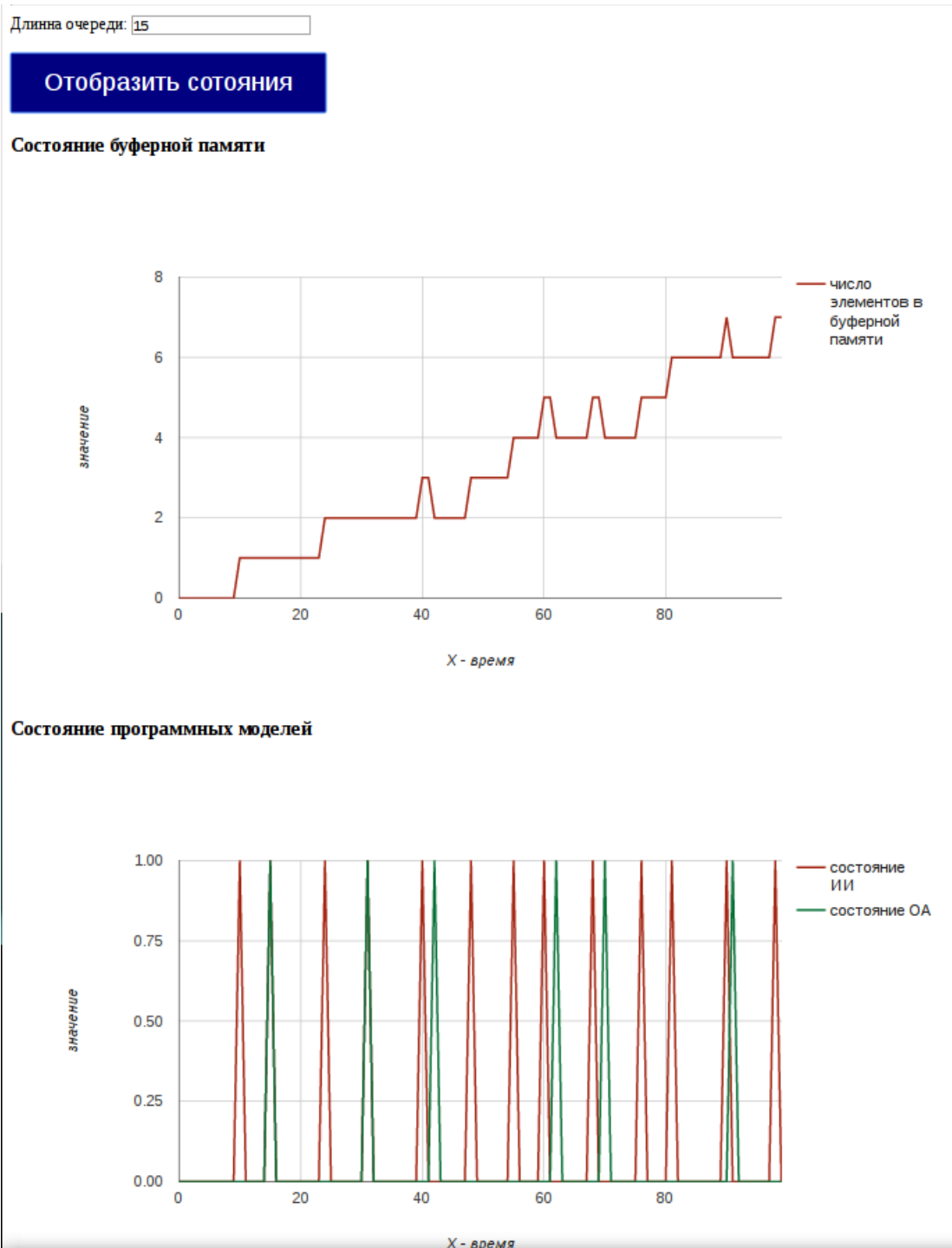


Рисунок 2.2 — Принцип  $\Delta t$ : графики

## Управляющая программа имитационной модели по событийному принципу

### Параметры имитационной модели

время работы:

☒ Выпускать обработанные заявки

### Источник информации

параметры равномерного распределения:

a:

b:

### Обслуживающий аппарат

параметры распределения Пуассона:

lambda:

Минимальная длина буферной памяти:

найти минимальную длину буферной памяти

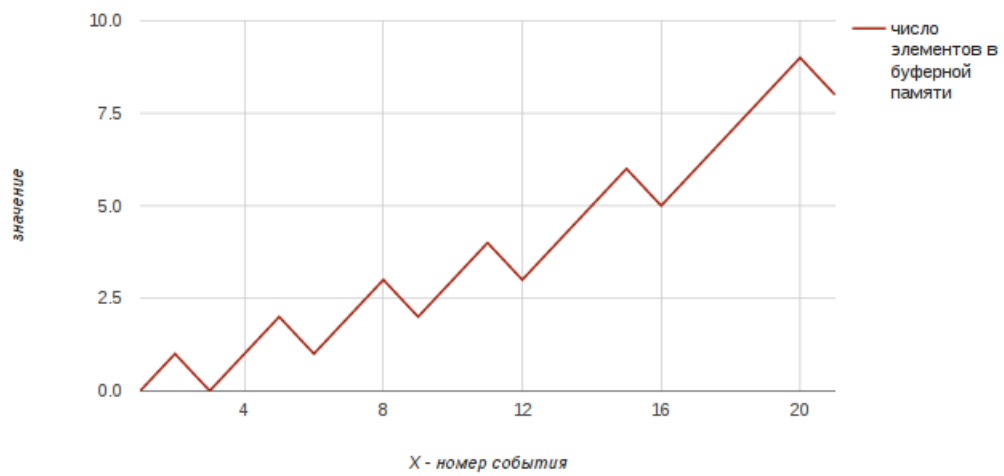
Рисунок 2.3 — Принцип событийный: поиск минимального размера буферной памяти



Длина очереди:

Отобразить состояния

Состояние буферной памяти



Состояние списка будущих событий

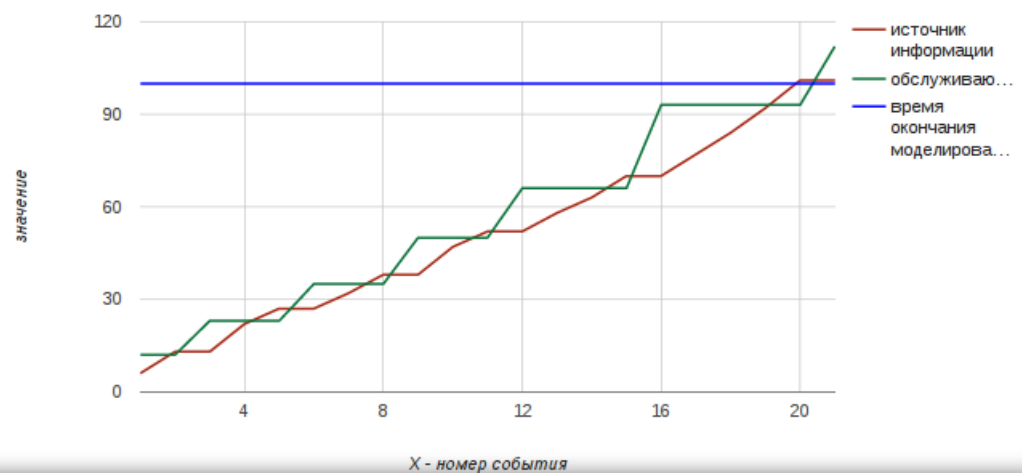


Рисунок 2.4 — Принцип событийный: графики