

# Лабораторная работа № 11

Модель системы массового обслуживания  $M|M|1$

Джахангиров Илгар Залид оглы

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Постановка задачи</b>	<b>6</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

3.1	Граф сети системы обработки заявок в очереди . . . . .	7
3.2	Граф генератора заявок системы . . . . .	8
3.3	Граф процесса обработки заявок на сервере системы . . . . .	8
3.4	Задание деклараций системы . . . . .	10
3.5	Параметры элементов основного графа системы обработки заявок в очереди . . . . .	11
3.6	Параметры элементов генератора заявок системы . . . . .	12
3.7	Параметры элементов обработчика заявок системы . . . . .	13
3.8	График изменения задержки в очереди . . . . .	14

# 1 Цель работы

Реализовать модель  $M|M|1$  в CPN tools.

## Задание

- Реализовать в CPN Tools модель системы массового обслуживания  $M|M|1$ .
- Настроить мониторинг параметров моделируемой системы и нарисовать графики очереди.

## **2 Выполнение лабораторной работы**

### 3 Постановка задачи

В систему поступает поток заявок двух типов, распределённый по пуассоновскому закону. Заявки поступают в очередь сервера на обработку. Дисциплина очереди - FIFO. Если сервер находится в режиме ожидания (нет заявок на сервере), то заявка поступает на обработку сервером.

Будем использовать три отдельных листа: на первом листе опишем граф системы (рис. ??), на втором — генератор заявок (рис. ??), на третьем — сервер обработки заявок (рис. ??).

Сеть имеет 2 позиции (очередь — Queue, обслуженные заявки — Complited) и два перехода (генерировать заявку — Arrivals, передать заявку на обработку серверу — Server). Переходы имеют сложную иерархическую структуру, задаваемую на отдельных листах модели (с помощью соответствующего инструмента меню — Hierarchy).

Между переходом Arrivals и позицией Queue, а также между позицией Queue и переходом Server установлена дуплексная связь. Между переходом Server и позицией Complited — односторонняя связь.

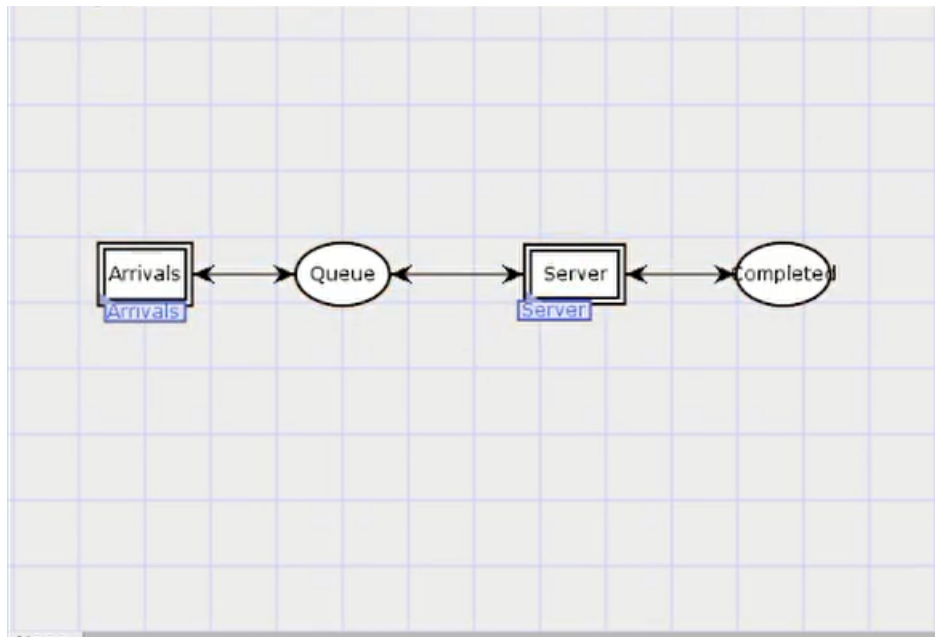


Рис. 3.1: Граф сети системы обработки заявок в очереди

Граф генератора заявок имеет 3 позиции (текущая заявка — Init, следующая заявка — Next, очередь — Queue из листа System) и 2 перехода (Init — определяет распределение поступления заявок по экспоненциальному закону с интенсивностью 100 заявок в единицу времени, Arrive — определяет поступление заявок в очередь).

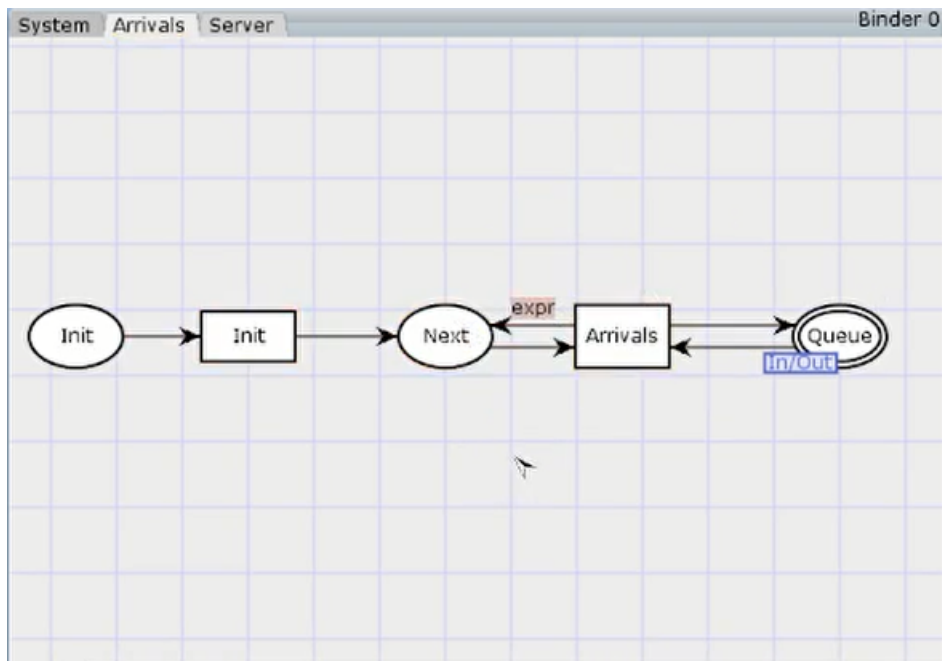


Рис. 3.2: Граф генератора заявок системы

Граф процесса обработки заявок на сервере имеет 4 позиции (Busy — сервер занят, Idle — сервер в режиме ожидания, Queue и Completed из листа System) и 2 перехода (Start — начать обработку заявки, Stop — закончить обработку заявки).

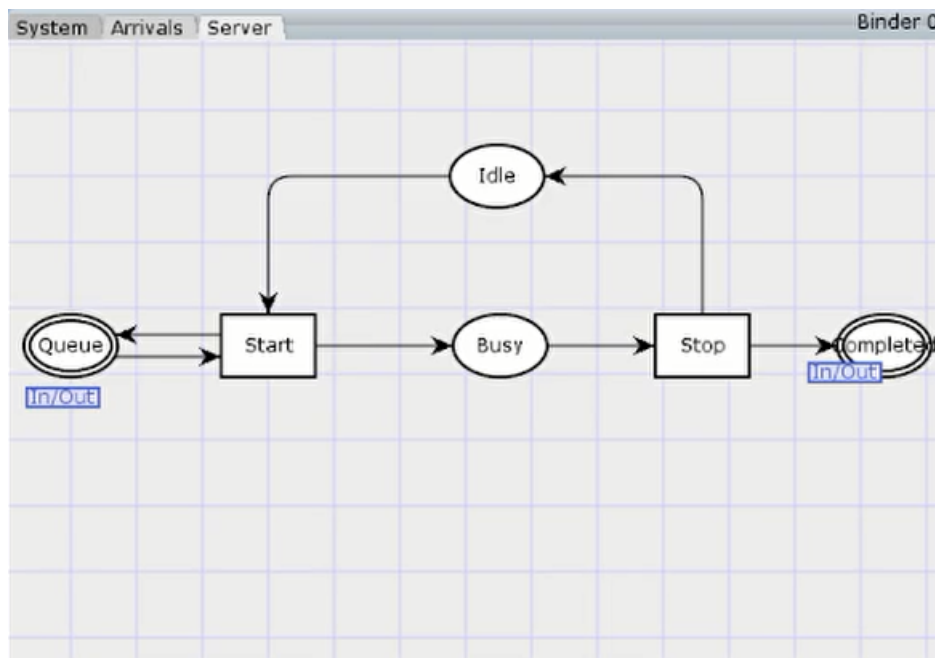


Рис. 3.3: Граф процесса обработки заявок на сервере системы



Зададим декларации системы (рис. ??).

Определим множества цветов системы (colorset):

- фишки типа UNIT определяют моменты времени;
- фишки типа INT определяют моменты поступления заявок в систему.
- фишки типа JobType определяют 2 типа заявок — А и В;
- кортеж Job имеет 2 поля: jobType определяет тип работы (соответственно имеет тип JobType, поле AT имеет тип INT и используется для хранения времени нахождения заявки в системе);
- фишки Jobs — список заявок;
- фишки типа ServerxJob — определяют состояние сервера, занятого обработкой заявок.

Переменные модели:

- proctime — определяет время обработки заявки;
- job — определяет тип заявки;
- jobs — определяет поступление заявок в очередь.

Определим функции системы:

- функция expTime описывает генерацию целочисленных значений через интервалы времени, распределённые по экспоненциальному закону;
- функция intTime преобразует текущее модельное время в целое число;
- функция newJob возвращает значение из набора Job — случайный выбор типа заявки (А или В).

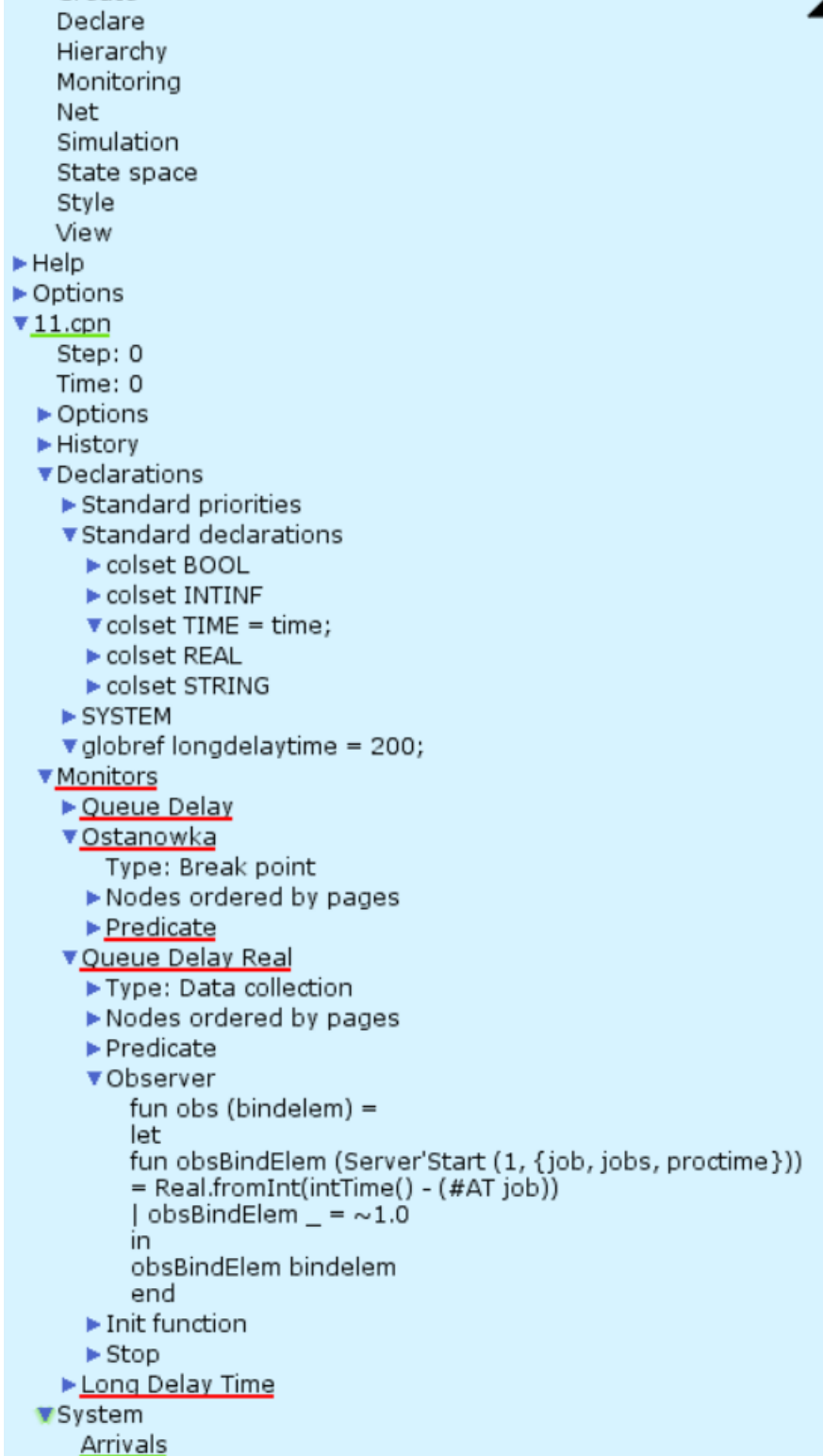


Рис. 3.4: Задание деклараций системы

Зададим параметры модели на графах сети.

На листе System (рис. 3.5):

- у позиции Queue множество цветов фишек — Jobs; начальная маркировка 1[] определяет, что изначально очередь пуста.
- у позиции Completed множество цветов фишек — Job.

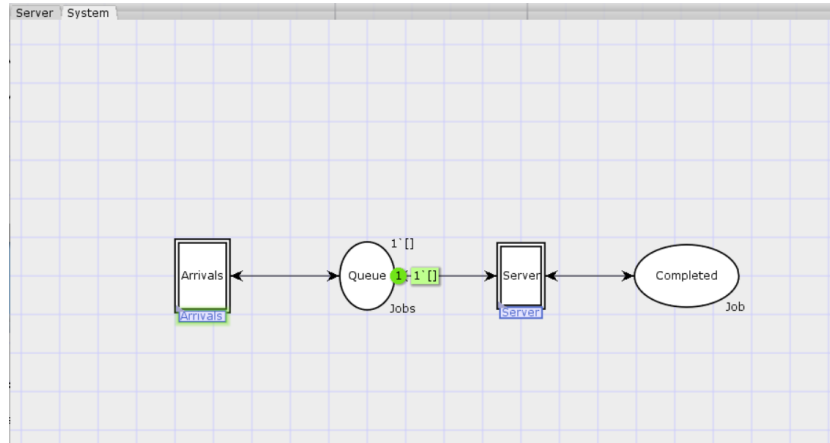


Рис. 3.5: Параметры элементов основного графа системы обработки заявок в очереди

На листе Arrivals (рис. ??):

- у позиции Init: множество цветов фишек — UNIT; начальная маркировка 1``()@0 определяет, что поступление заявок в систему начинается с нулевого момента времени;
- у позиции Next: множество цветов фишек — UNIT;
- на дуге от позиции Init к переходу Init выражение () задаёт генерацию заявок;
- на дуге от переходов Init и Arrive к позиции Next выражение ()@+expTime(100) задаёт экспоненциальное распределение времени между поступлениями заявок;
- на дуге от позиции Next к переходу Arrive выражение () задаёт перемещение фишки;

- на дуге от перехода Arrive к позиции Queue выражение  $jobs^{^^}[job]$  задает поступление заявки в очередь;
- на дуге от позиции Queue к переходу Arrive выражение  $jobs$  задаёт обратную связь.

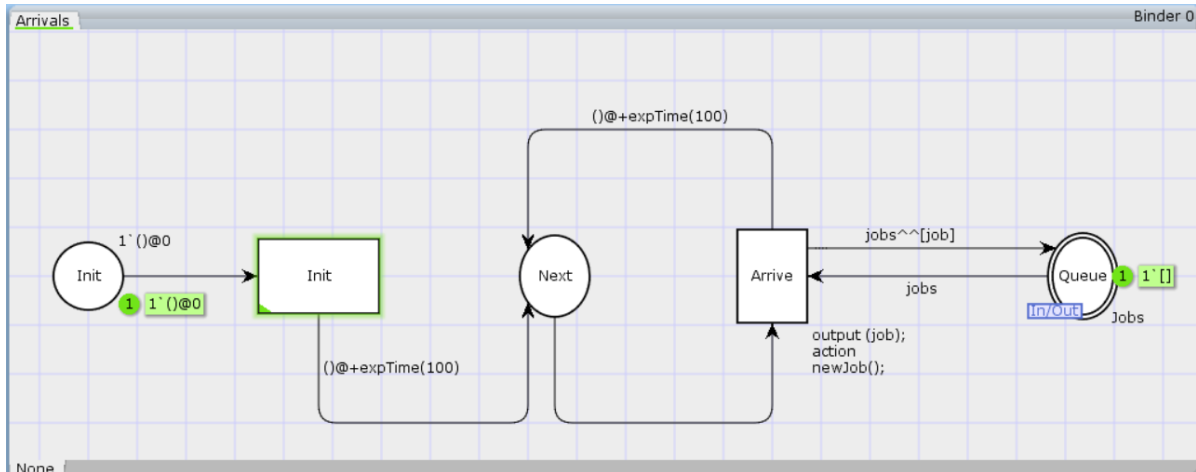


Рис. 3.6: Параметры элементов генератора заявок системы

На листе Server (рис. ??):

- у позиции Busy: множество цветов фишек — Server, начальное значение маркировки —  $1'`server@0$  определяет, что изначально на сервере нет заявок на обслуживание;
- у позиции Idle: множество цветов фишек —  $Server \times Job$ ;
- переход Start имеет сегмент кода `output (proctime); action expTime(90);` определяющий, что время обслуживания заявки распределено по экспоненциальному закону со средним временем обработки в 90 единиц времени;
- на дуге от позиции Queue к переходу Start выражение  $job::jobs$  определяет, что сервер может начать обработку заявки, если в очереди есть хотя бы одна заявка;
- на дуге от перехода Start к позиции Busy выражение  $(server, job)@+proctime$  запускает функцию расчёта времени обработки заявки на сервере;

- на дуге от позиции Busy к переходу Stop выражение `(server, job)` говорит о завершении обработки заявки на сервере;
- на дуге от перехода Stop к позиции Completed выражение `job` показывает, что заявка считается обслуженной;
- выражение `server` на дугах от и к позиции Idle определяет изменение состояние сервера (обрабатывает заявки или ожидает);
- на дуге от перехода Start к позиции Queue выражение `jobs` задаёт обратную связь.

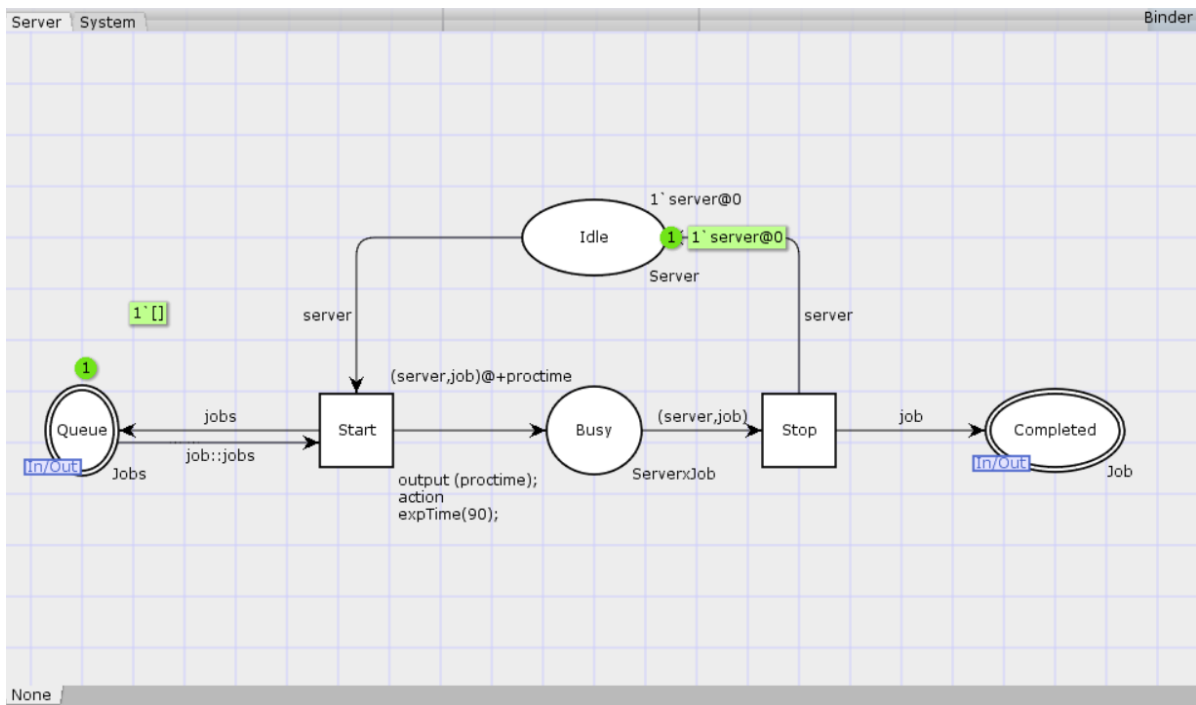


Рис. 3.7: Параметры элементов обработчика заявок системы

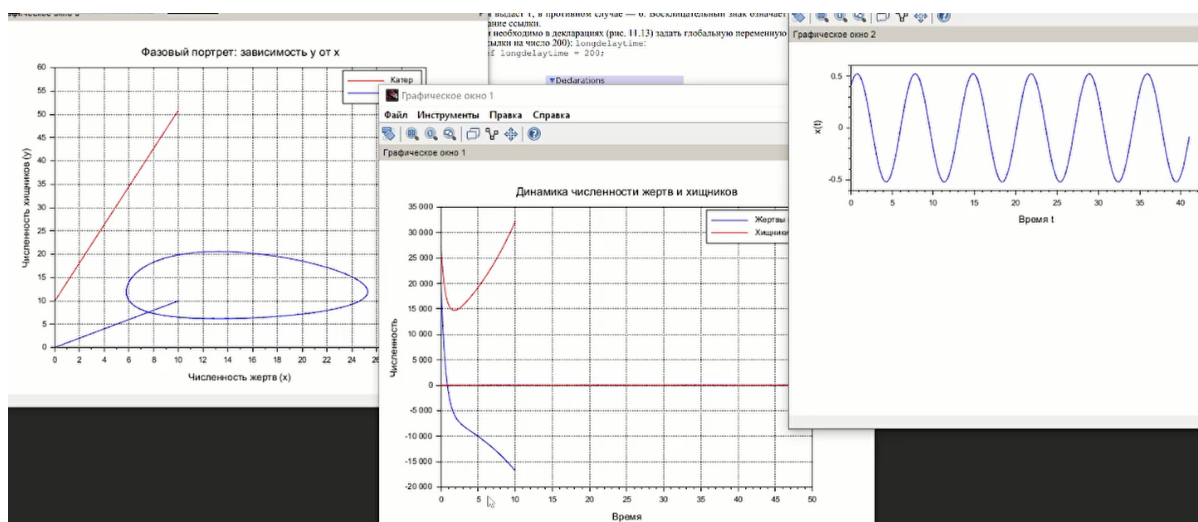


Рис. 3.8: График изменения задержки в очереди

## 4 Выводы

В процессе выполнения данной лабораторной работы я реализовал модель системы массового обслуживания  $M|M|1$  в CPN Tools.