

Отчёт по лабораторной работе №11

Модель системы массового обслуживания $M|M|1$

Ощепков Дмитрий Владимирович НФИбд-01-22

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 14 |

Список иллюстраций

| | | |
|-----|---|----|
| 3.1 | граф системы | 7 |
| 3.2 | генератор заявок | 8 |
| 3.3 | сервер обработки заявок | 8 |
| 3.4 | Параметры элементов основного графа системы обработки заявок в очереди | 9 |
| 3.5 | Параметры элементов генератора заявок системы | 10 |
| 3.6 | Параметры элементов обработчика заявок системы | 10 |
| 3.7 | Запуск системы обработки заявок в очереди | 11 |
| 3.8 | График изменения задержки в очереди | 12 |
| 3.9 | Периоды времени, когда значения задержки в очереди превышали заданное значение | 13 |

Список таблиц

1 Цель работы

Реализовать Модель системы массового обслуживания $M|M|1$

2 Задание

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.

3 Выполнение лабораторной работы

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

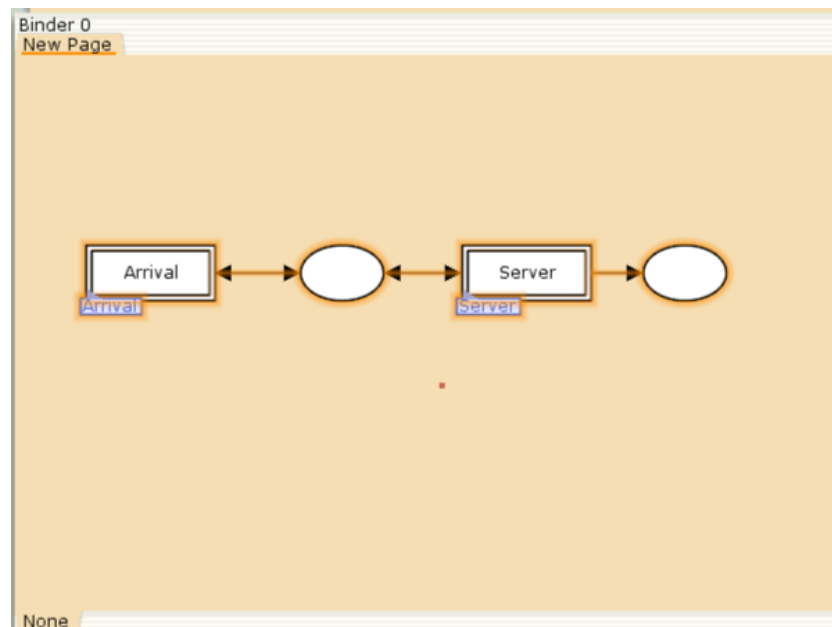


Рис. 3.1: граф системы

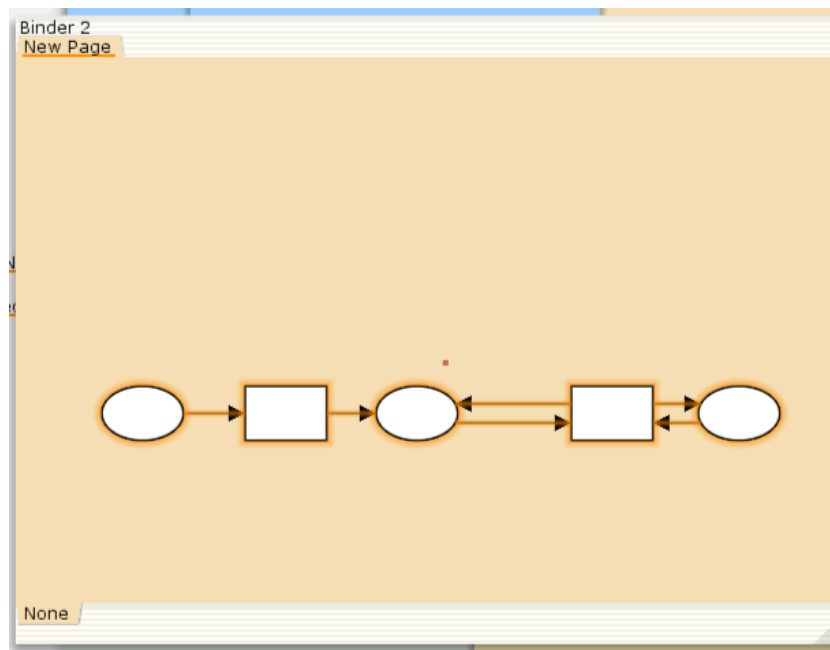


Рис. 3.2: генератор заявок

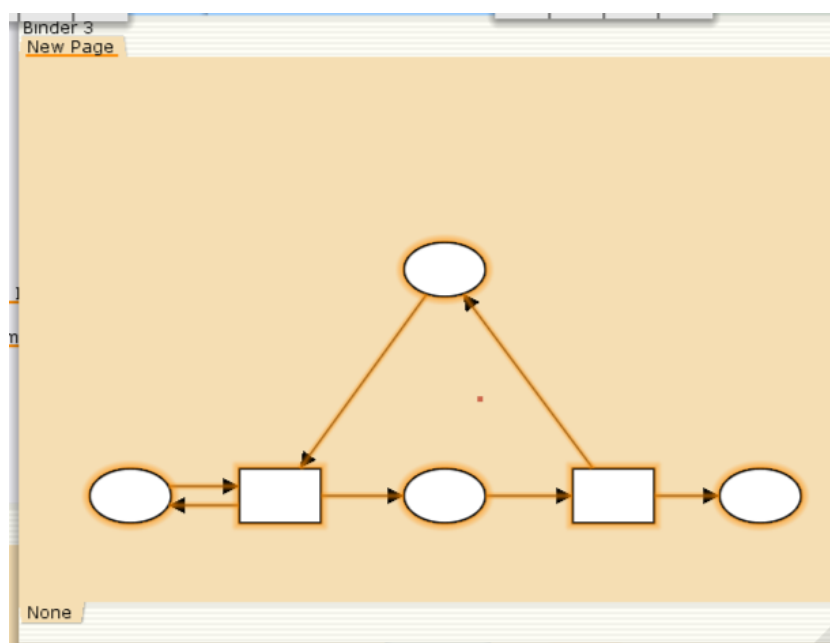


Рис. 3.3: сервер обработки заявок

Определим декларации

Time: 0

- Options
- History
- Declarations
 - Standard declarations
 - colset UNIT = unit timed;
 - colset INT = int;
 - colset Server = with server timed;
 - colset JobType = with A | B;
 - colset Job = record jobType: JobType * AT : INT;
 - colset Jobs = list Job;
 - colset ServerxJob = product Server * Job timed;
 - var proctime: INT;
 - var job: Job;
 - var jobs: jobs;
 - fun newJob() = {jobType = JobType.ran(), AT = intTime()};
 - fun intTime() = IntInf.toInt(time());
 - fun expTime(mean: int) =
 let
 val realMean = Real.fromInt mean
 val rv = exponential((1.0/realMean))
 in
 floor(rv+0.5)
 end;
- Monitors
- New Page
 - Arrival
 - Server
- New net.cpn
 - Step: 0
 - Time: 0

Binder 2
New Page

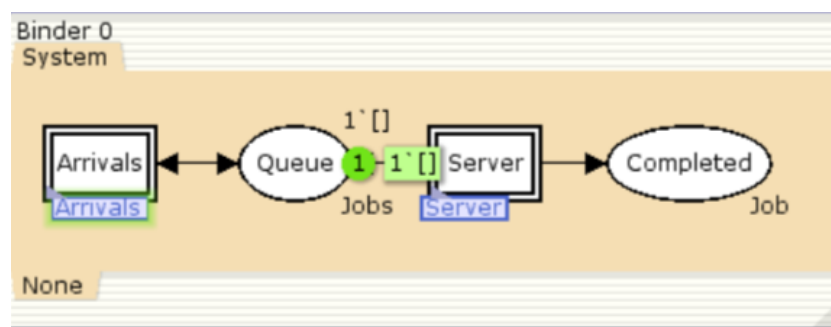


Рис. 3.4: Параметры элементов основного графа системы обработки заявок в очереди

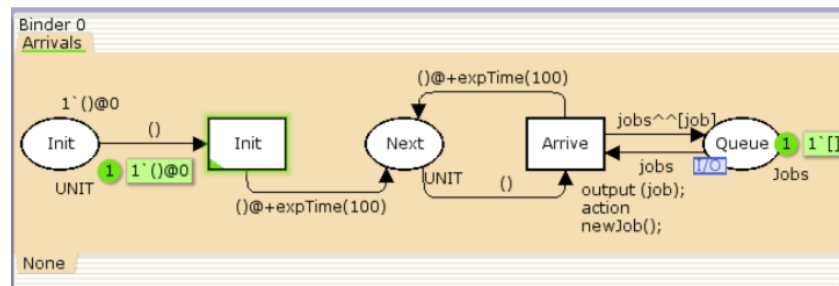


Рис. 3.5: Параметры элементов генератора заявок системы

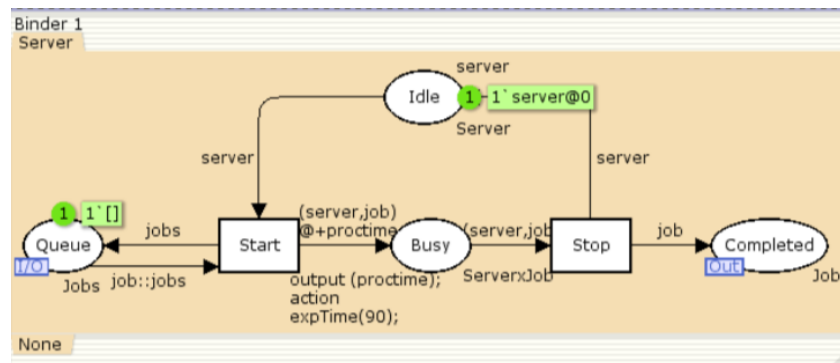


Рис. 3.6: Параметры элементов обработчика заявок системы

Мониторинг параметров моделируемой системы

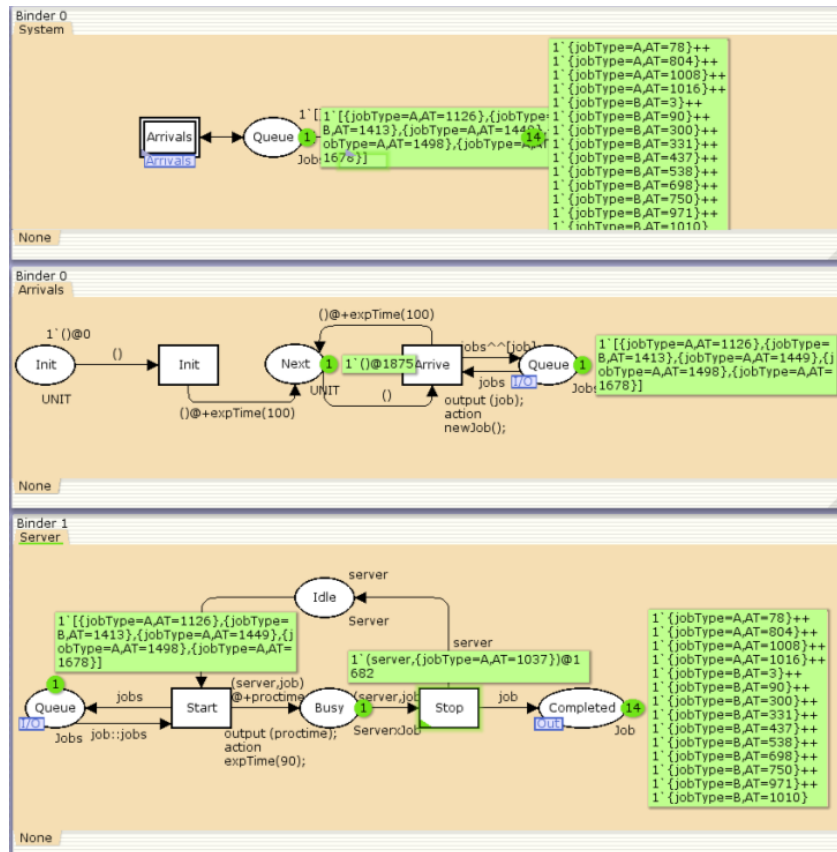


Рис. 3.7: Запуск системы обработки заявок в очереди

Добавили функцию

```
fun obs (bindelem) = let fun obsBindElem (Server'Start (1, {job, jobs, proctime})) =
(intTime() - (#AT job)) | obsBindElem _ = ~1 in obsBindElem bindelem end
```

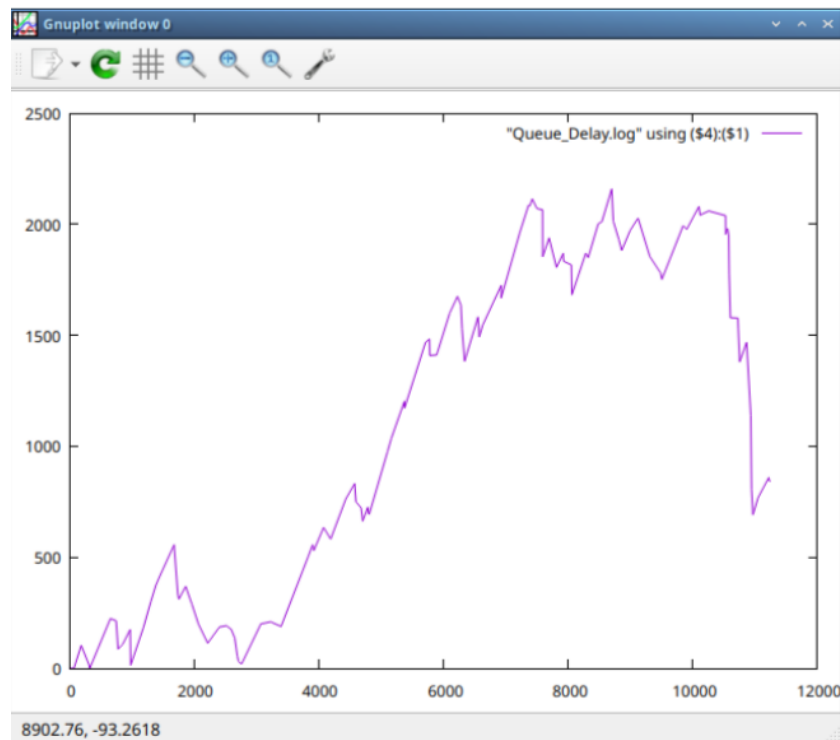


Рис. 3.8: График изменения задержки в очереди

Функцию Observer изменим следующим образом: `fun obs (bindelem) = if IntInf.tilnt(Queue_Delay.last())>=(!longdelaytime) then 1 else 0`

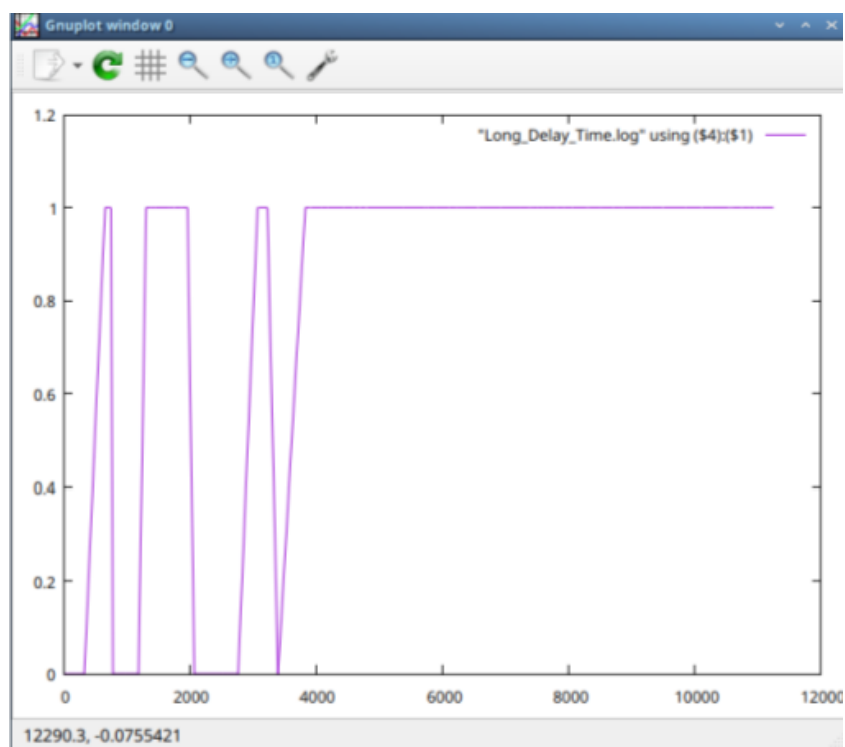


Рис. 3.9: Периоды времени, когда значения задержки в очереди превышали заданное значение

4 Выводы

Реализовал Модель системы массового обслуживания $M|M|1$