

Лабораторная работа 11

Имитационное моделирование

Голощапов Ярослав Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16

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	Декларации	9
3.5	Симуляция	10
3.6	Ostanovka и Queue_Delay	10
3.7	Симуляция	11
3.8	Queue_Delay	11
3.9	Код	12
3.10	График	12
3.11	Queue Delay Real	13
3.12	Queue Delay Real	13
3.13	Long Delay Time	14
3.14	Long Delay Time	14
3.15	График	15

Список таблиц

1 Цель работы

Построение модели 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

1. Будем использовать три отдельных листа: на первом листе опишем граф системы (рис. 3.1) , на втором — генератор заявок (рис. 3.2), на третьем — сервер обработки заявок (рис. 3.3). 1.1 Сеть имеет 2 позиции (очередь — Queue, обслуженные заявки — Completed) 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).

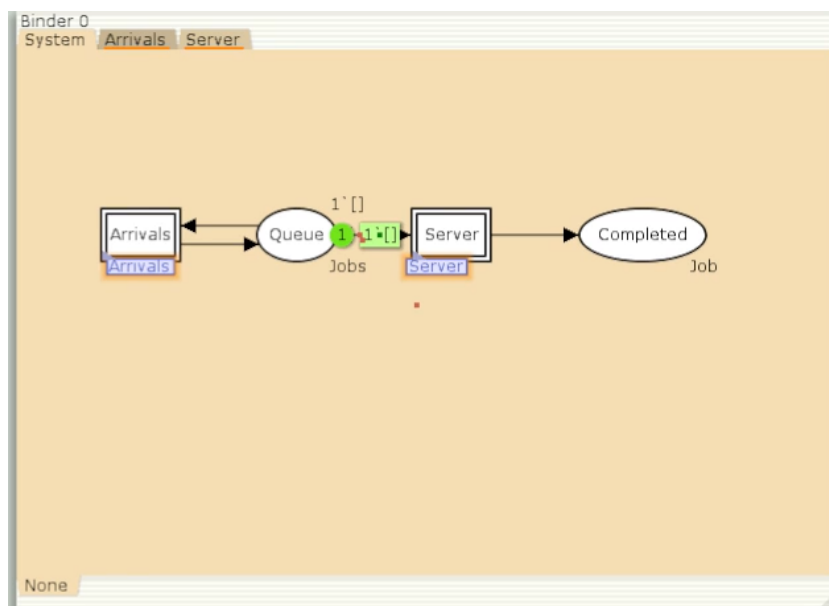


Рис. 3.1: Граф сети системы обработки заявок в очереди

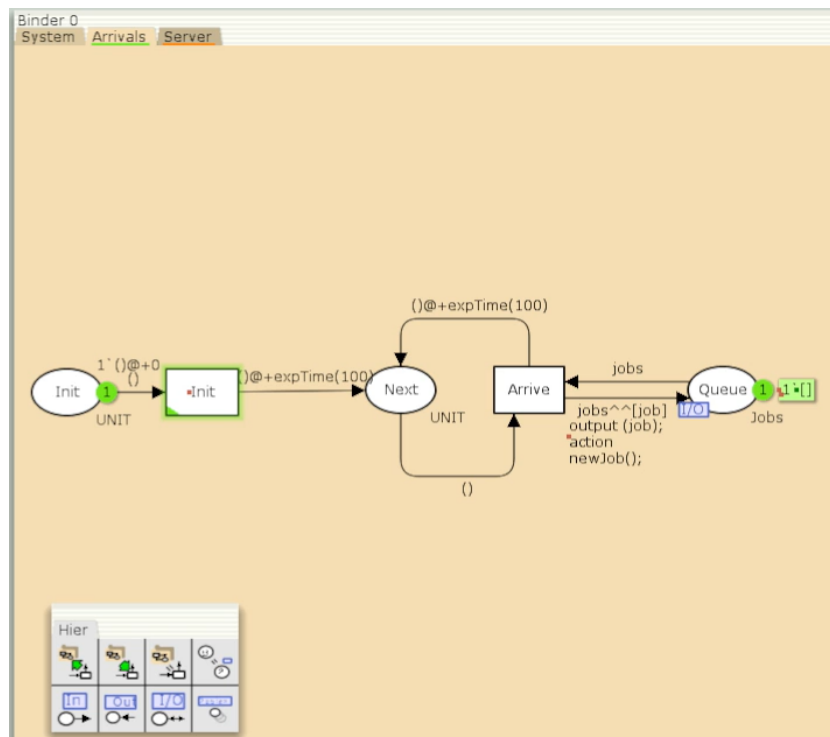


Рис. 3.2: Граф генератора заявок системы

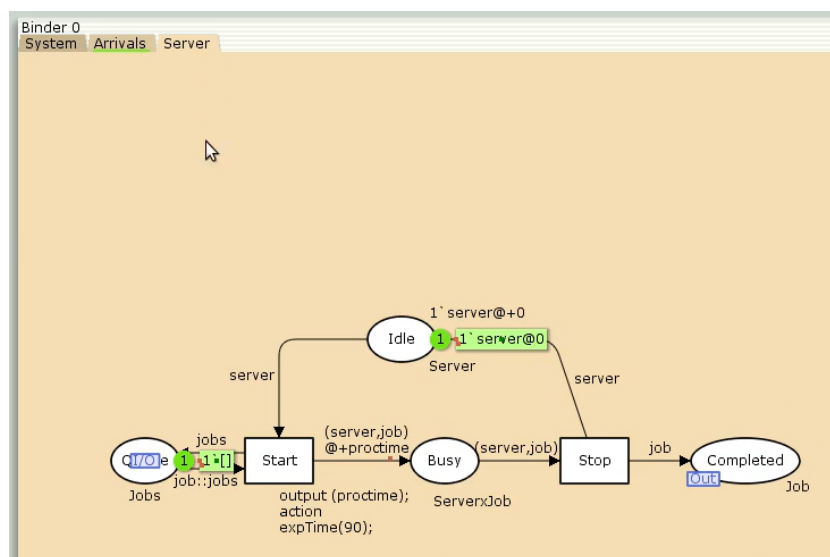


Рис. 3.3: Граф процесса обработки заявок на сервере системы

Записали новые декларации (рис. 3.4).


```

▼ petry_qm.cpn
  Step: 0
  Time: 0
  ► Options
  ► History
  ▼ Declarations
    ► Standard declarations
    ▼ System
      ▼ colset INT = int;
      ▼ colset UNIT = unit timed;
      ▼ 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 expTime (mean: int) =
        let
          val realMean = Real.fromInt mean
          val rv = exponential ((1.0/realMean))
        in
          floor(rv+0.5)
        end;
      ▼ fun intTime() = IntInf.toInt(time());
      ▼ fun newJob() = {jobType = JobType.ran(), AT = intTime()};
    ▼ Monitors
    ▼ System
      Arrivals
      Server

```

Рис. 3.4: Декларации

Запускаем симуляцию(рис. 3.5)

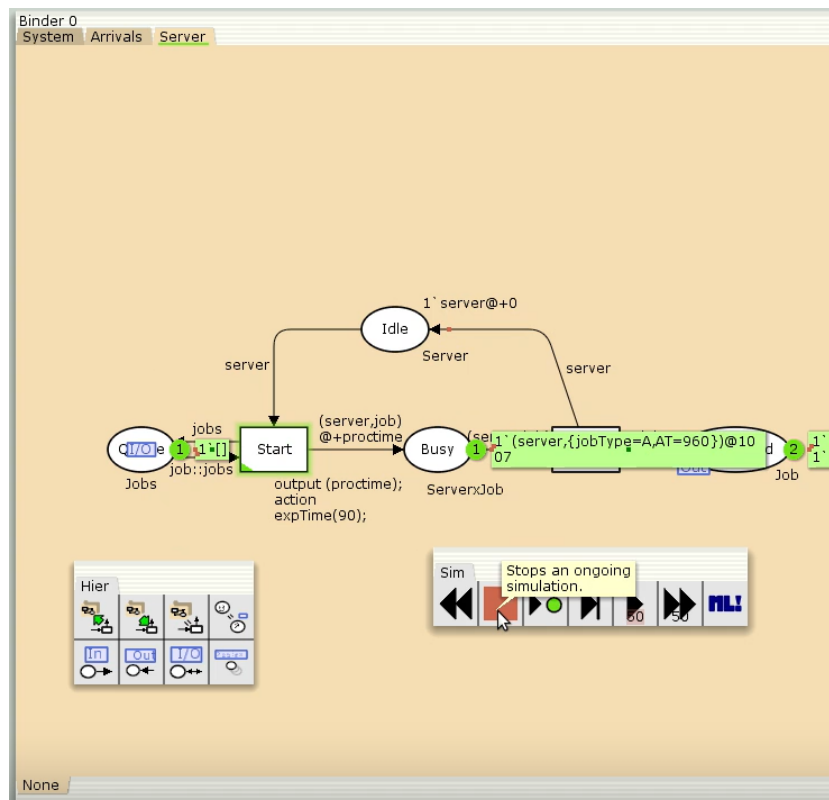


Рис. 3.5: Симуляция

Создаем 2 новых подраздела в Monitor - Ostanovka и Queue_Delay (рис. 3.6)



Рис. 3.6: Ostanovka и Queue_Delay

Запуск симуляции с новыми подразделами (рис. 3.7)

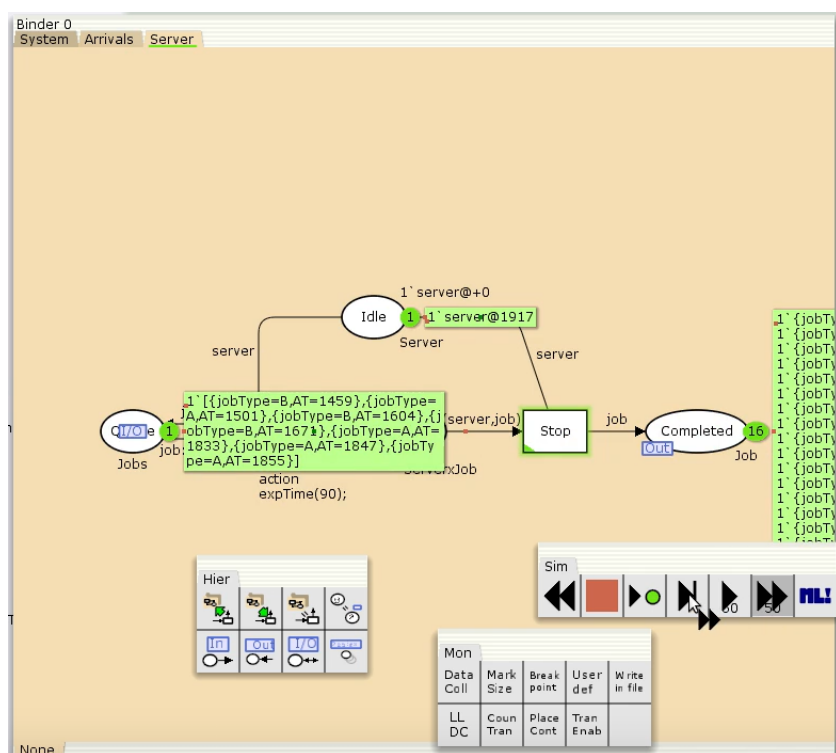


Рис. 3.7: Симуляция

Вывод с файла Queue_Delay(рис. 3.8).

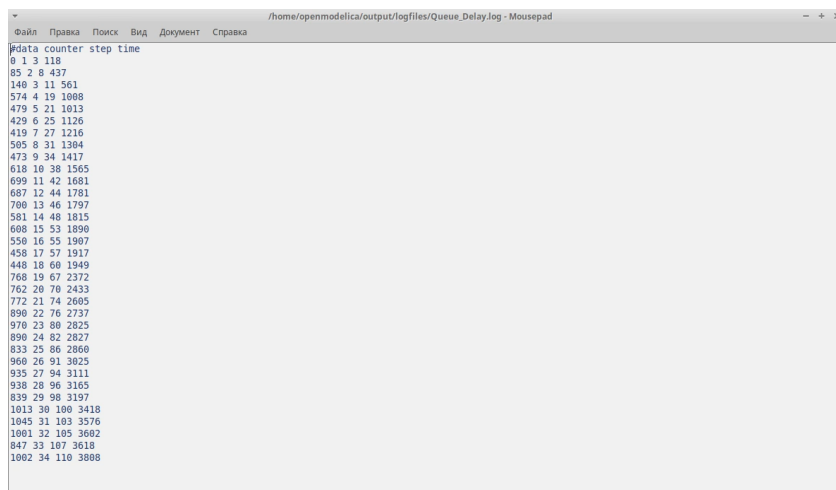


Рис. 3.8: Queue_Delay

Код в gnuplot (рис. 3.9).

```

/home/openmodelica/output/logfiles/graph_plot - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка

#!/usr/bin/gnuplot -persist

set encoding utf8
set term pngcairo font "Helvetica,9"

set out 'window_1.png'
plot "Queue_Delay.log" using ($4):($1) with lines

```

Рис. 3.9: Код

Вывод графика в gnuplot (рис. 3.10)

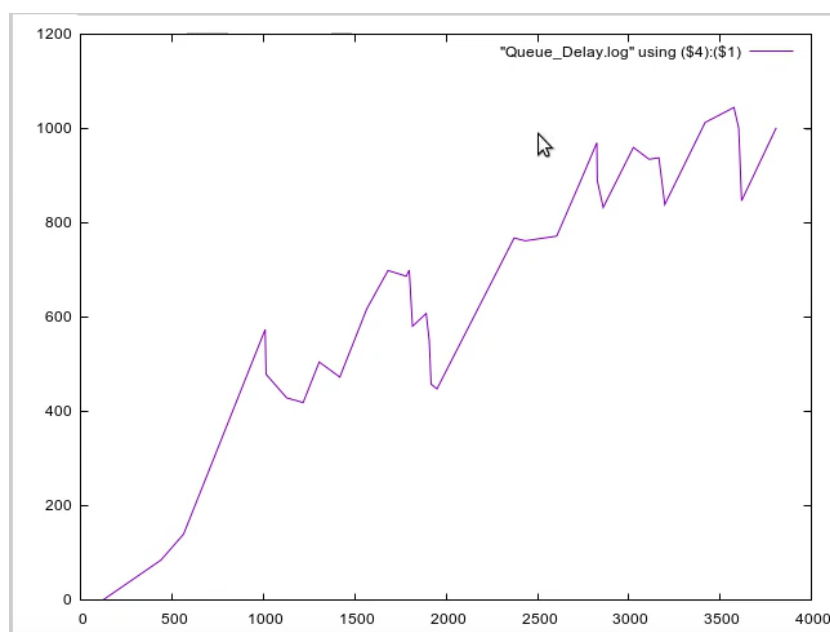


Рис. 3.10: График

Добавили новый подраздел Queue Delay Real (рис. 3.11)

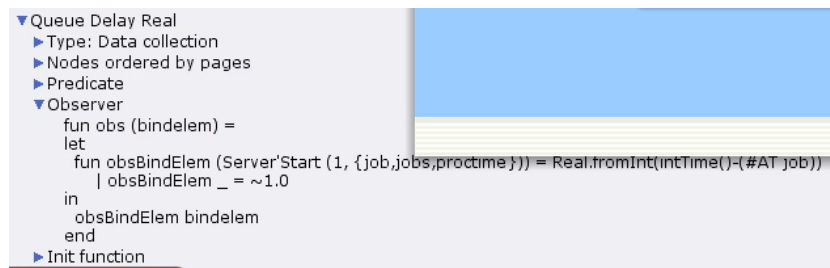


Рис. 3.11: Queue Delay Real

Содержимое файла Queue Delay Real (рис. 3.12)

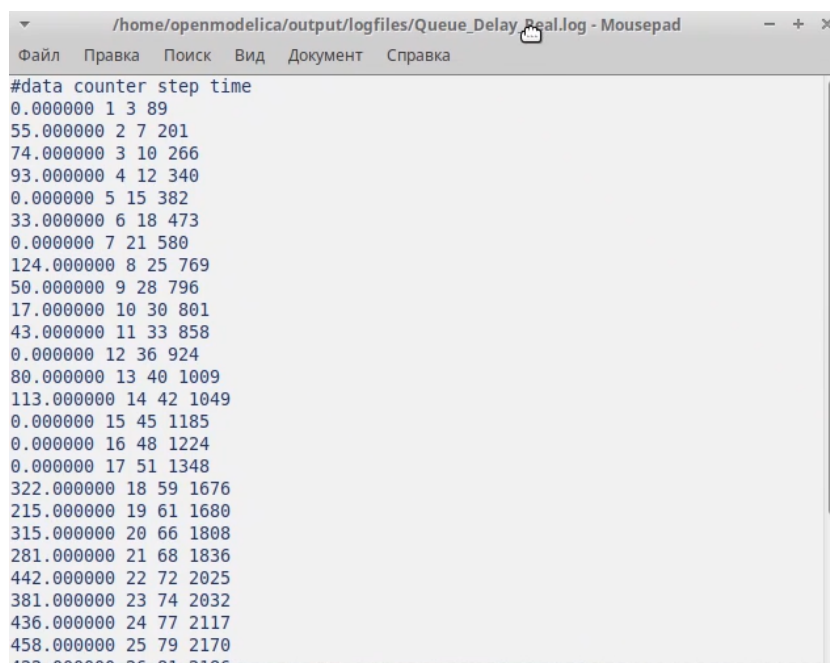


Рис. 3.12: Queue Delay Real

Добавили новый подраздел Long Delay Time (рис. 3.13)

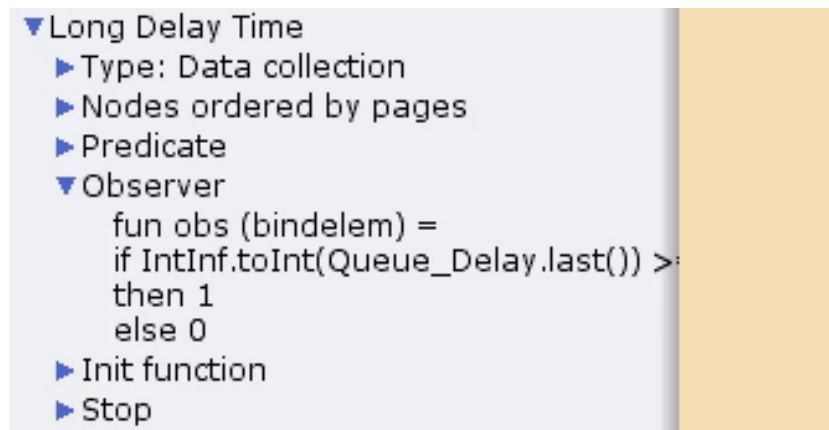


Рис. 3.13: Long Delay Time

Содержимое файла Long Delay Time (рис. 3.14)

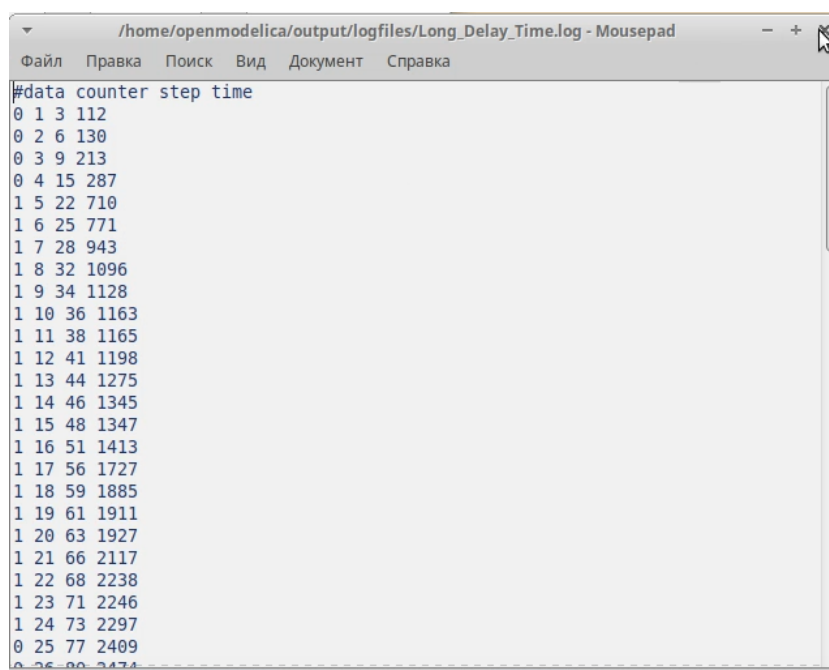


Рис. 3.14: Long Delay Time

Изменили немного код и вывели график в gnuplot(рис. 3.15)

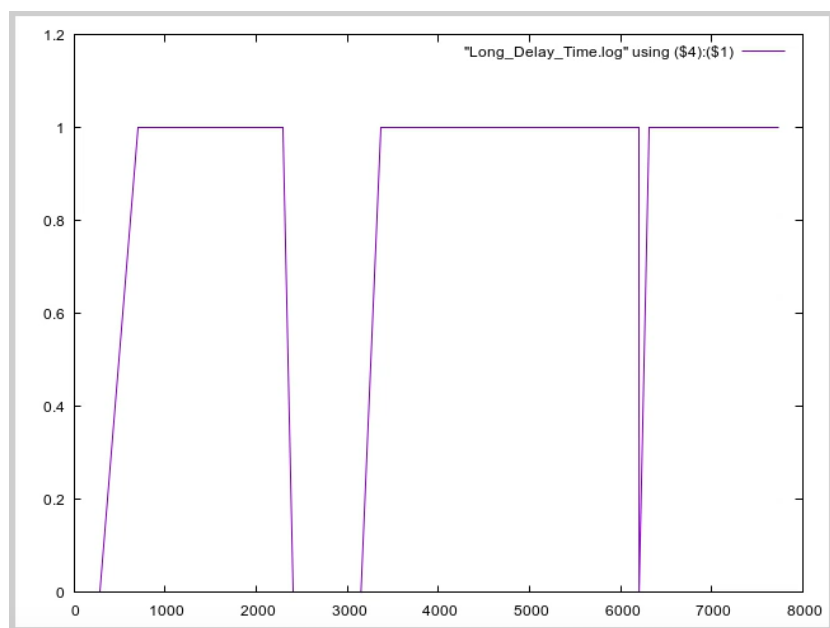


Рис. 3.15: График

4 Выводы

В этой лабораторной работе я приобрел навыки построения модели системы массового обслуживания $M | M | 1$