

Лабораторная работа 11

Имитационное моделирование

Голощапов Ярослав Вячеславович

10 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Голощапов Ярослав Вячеславович
- студент 3 курса
- Российский университет дружбы народов
- 1132222003@pfur.ru
- <https://yvgoloschapov.github.io/ru/>

Построение модели системы массового обслуживания $M | M | 1$

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.

Выполнение лабораторной работы

1. Будем использовать три отдельных листа: на первом листе опишем граф системы , на втором — генератор заявок , на третьем — сервер обработки заявок . 1.1 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).

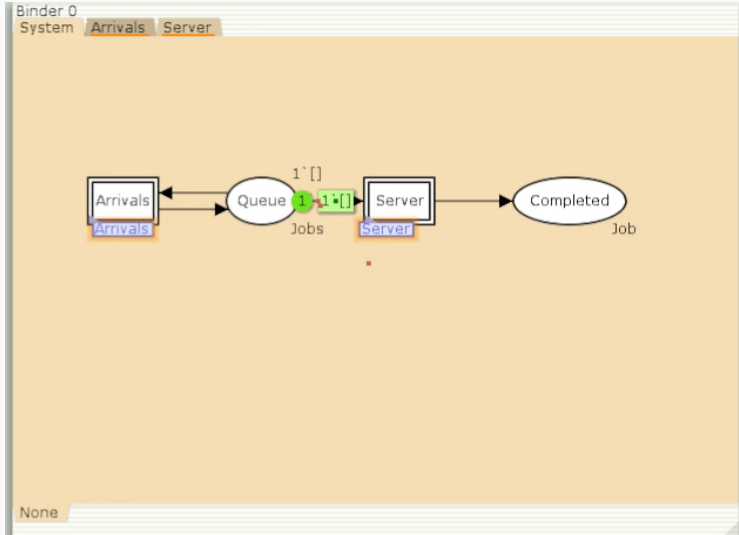


Рис. 1: Граф сети системы обработки заявок в очереди

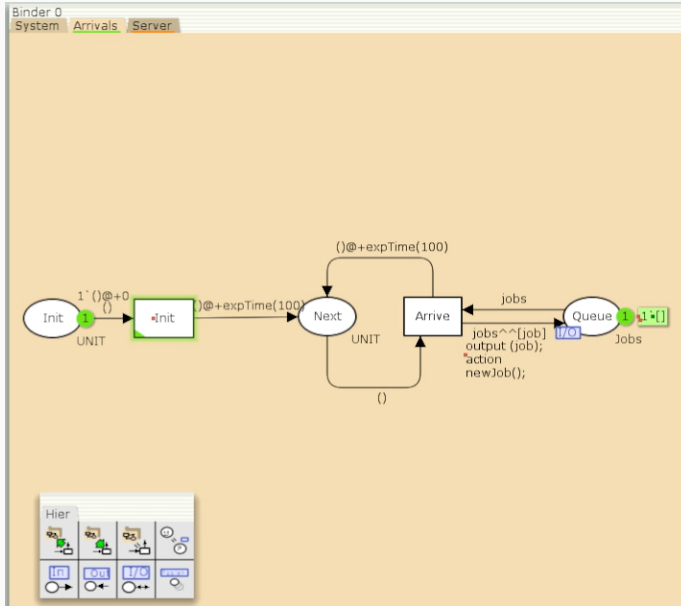


Рис. 2: Граф генератора заявок системы

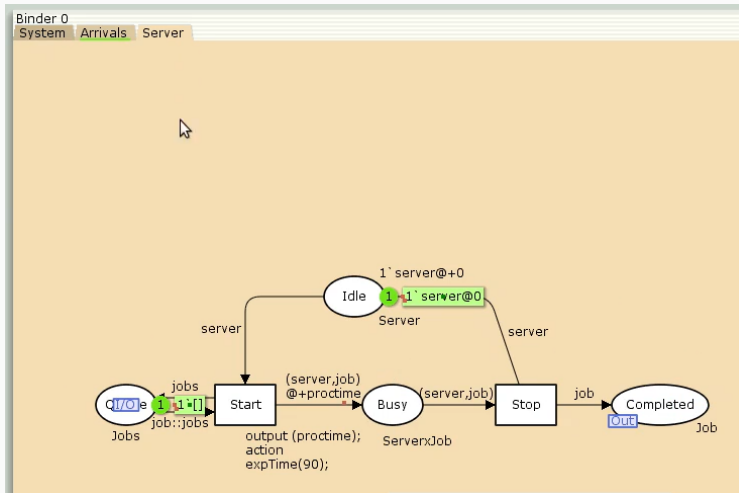
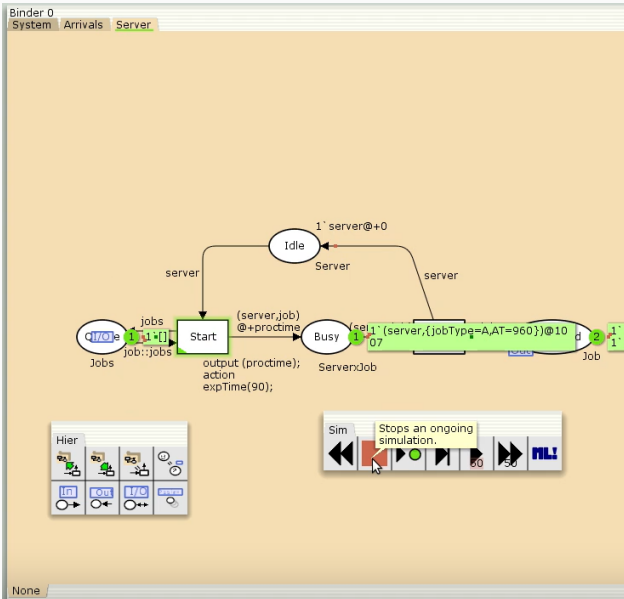


Рис. 3: Граф процесса обработки заявок на сервере системы

Записали новые декларации .

```
▼ 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
```

Запускаем симуляцию

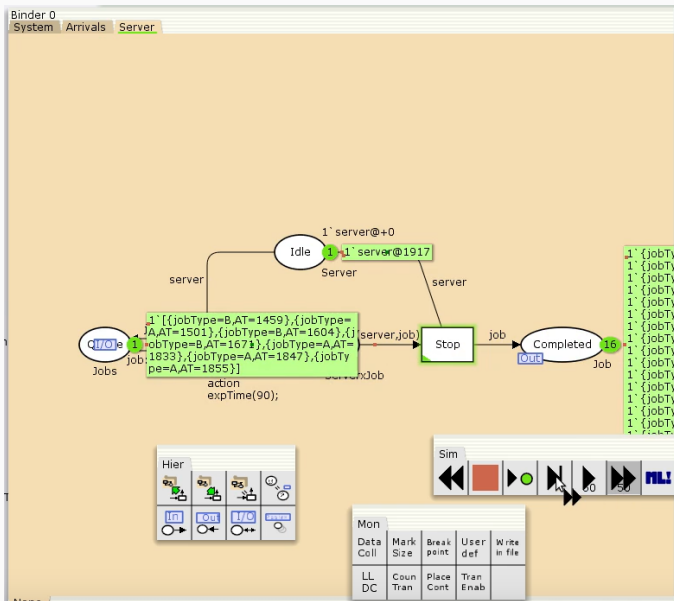


Создаем 2 новых подраздела в Monitor - Ostanovka и Queue_Delay

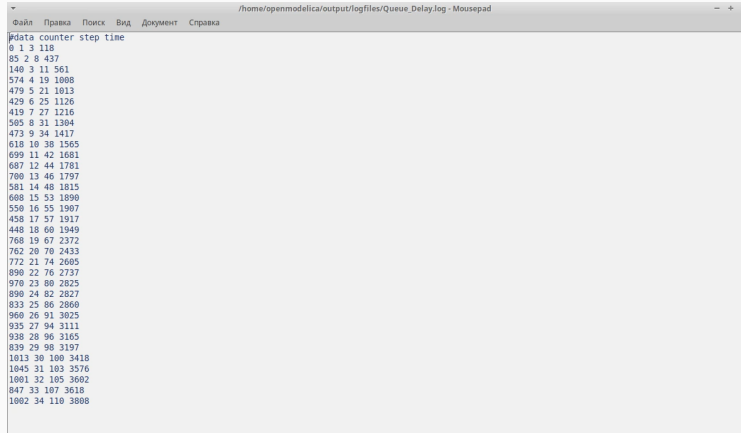
```
▼ Monitors
  ▼ Queue Delay
    ▶ Type: Data collection
    ▶ Nodes ordered by pages
    ▼ Predicate
      fun pred (bindelem) =
      let
        fun predBindElem (Server'Start (1,
                                     {job,jobs,proctime})) =
          | predBindElem _ = false
      in
        predBindElem bindelem
      end
    ▶ Observer
    ▶ Init function
    ▶ Stop
    ▶ Ostanovka
```

Рис. 6: Ostanovka и Queue_Delay

Запуск симуляции с новыми подразделами

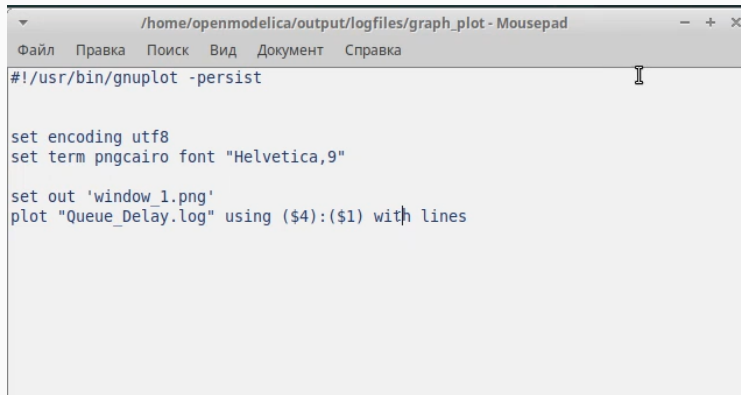


Вывод с файла Queue_Delay.



```
#data counter step time
0 1 3 118
85 2 8 437
140 3 11 561
574 4 19 1008
479 5 21 1013
429 6 25 1126
419 7 27 1216
505 8 31 1304
473 9 34 1417
618 10 38 1565
699 11 42 1681
687 12 44 1781
700 13 46 1797
581 14 48 1815
608 15 53 1890
550 16 55 1907
458 17 57 1917
448 18 60 1949
768 19 67 2372
762 20 70 2433
772 21 74 2605
890 22 76 2737
970 23 80 2825
890 24 82 2827
833 25 86 2860
960 26 91 3025
935 27 94 3111
938 28 96 3165
839 29 98 3197
1013 30 100 3418
1045 31 103 3576
1001 32 105 3602
847 33 107 3618
1002 34 110 3808
```

Рис. 8: Queue_Delay

A screenshot of a text editor window titled "/home/openmodelica/output/logfiles/graph_plot - Mousepad". The window has a menu bar with options: "Файл", "Правка", "Поиск", "Вид", "Документ", and "Справка". The text area contains the following gnuplot code:

```
#!/usr/bin/gnuplot -persist

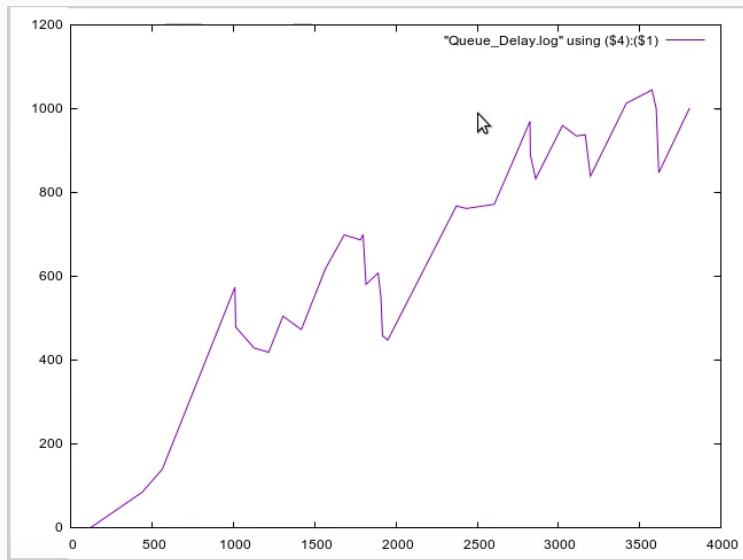
set encoding utf8
set term pngcairo font "Helvetica,9"

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

A cursor is visible at the end of the first line of code.

Рис. 9: Код

Вывод графика в gnuplot



Добавили новый подраздел Queue Delay Real

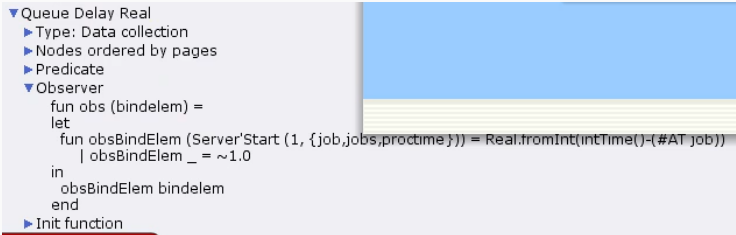


Рис. 11: Queue Delay Real

Содержимое файла Queue Delay Real

```

/home/openmodelica/output/logfiles/Queue_Delay_Real.log - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
#data counter step time
0.000000 1 3 89
55.000000 2 7 201
74.000000 3 10 266
93.000000 4 12 340
0.000000 5 15 382
33.000000 6 18 473
0.000000 7 21 580
124.000000 8 25 769
50.000000 9 28 796
17.000000 10 30 801
43.000000 11 33 858
0.000000 12 36 924
80.000000 13 40 1009
113.000000 14 42 1049
0.000000 15 45 1185
0.000000 16 48 1224
0.000000 17 51 1348
322.000000 18 59 1676
215.000000 19 61 1680
315.000000 20 66 1808
281.000000 21 68 1836
442.000000 22 72 2025
381.000000 23 74 2032
436.000000 24 77 2117
458.000000 25 79 2170
422.000000 26 81 2186

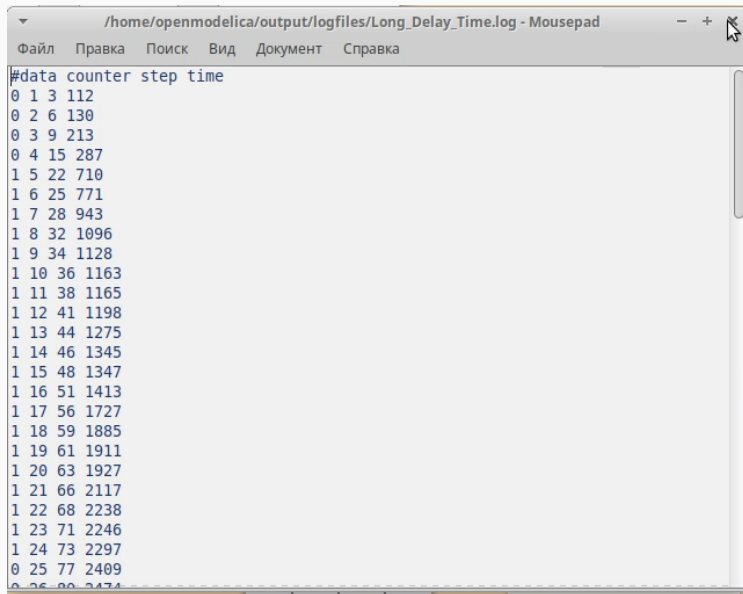
```

- ▼ Long Delay Time
 - ▶ Type: Data collection
 - ▶ Nodes ordered by pages
 - ▶ Predicate
 - ▼ Observer

```
fun obs (bindelem) =  
  if IntInf.toInt(Queue_Delay.last()) >:  
  then 1  
  else 0
```
 - ▶ Init function
 - ▶ Stop

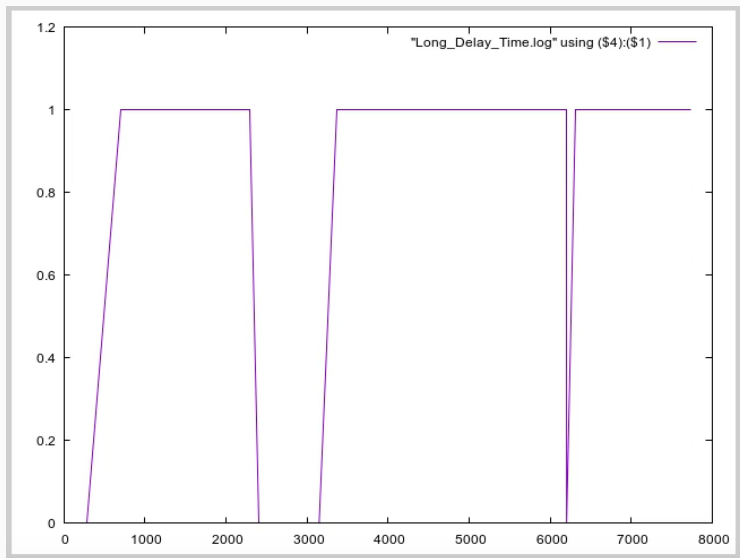
Рис. 13: Long Delay Time

Содержимое файла Long Delay Time

A screenshot of a window titled "/home/openmodelica/output/logfiles/Long_Delay_Time.log - Mousepad". The window has a menu bar with "Файл", "Правка", "Поиск", "Вид", "Документ", and "Справка". The main area contains a table of data with columns "#data", "counter", "step", and "time". The data is as follows:

#data	counter	step	time
0	1	3	112
0	2	6	130
0	3	9	213
0	4	15	287
1	5	22	710
1	6	25	771
1	7	28	943
1	8	32	1096
1	9	34	1128
1	10	36	1163
1	11	38	1165
1	12	41	1198
1	13	44	1275
1	14	46	1345
1	15	48	1347
1	16	51	1413
1	17	56	1727
1	18	59	1885
1	19	61	1911
1	20	63	1927
1	21	66	2117
1	22	68	2238
1	23	71	2246
1	24	73	2297
0	25	77	2409
0	26	80	2474

Изменили немного код и вывели график в gnuplot



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