# 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	0
3.6	Ostanovka и Queue_Delay	.0
3.7	Симуляция	.1
3.8	Queue_Delay	.1
3.9	Код	2
3.10	График	2
3.11	Queue Delay Real	.3
3.12	Queue Delay Real	.3
3.13	Long Delay Time	4
3.14	Long Delay Time	4
3.15	График	5

# Список таблиц

# 1 Цель работы

Построение модели системы массового обслуживания М  $\mid$ М  $\mid$ 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 — Complited) и два перехода (генерировать заявку — Arrivals, передать заявку на обработку сер- веру — Server). Переходы имеют сложную иерархическую структуру, задаваемую на отдельных листах модели (с помощью соответствующего инструмента меню — Hierarchy).

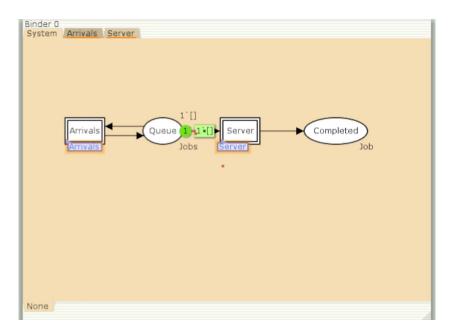


Рис. 3.1: Граф сети системы обработки заявок в очереди

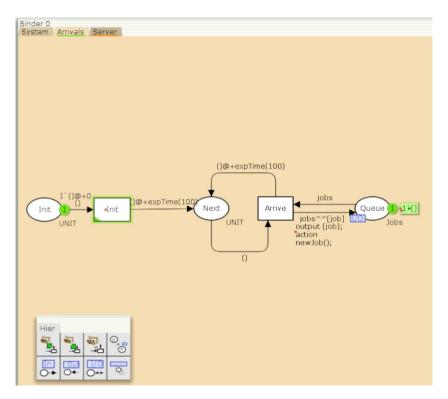


Рис. 3.2: Граф генератора заявок системы

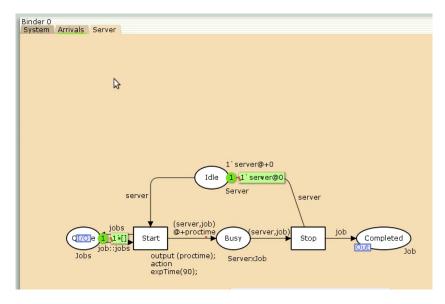


Рис. 3.3: Граф процесса обработки заявок на сервере системы

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) =
       val realMean = Real.fromInt mean
       val rv = exponential ((1.0/realMean))
       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)

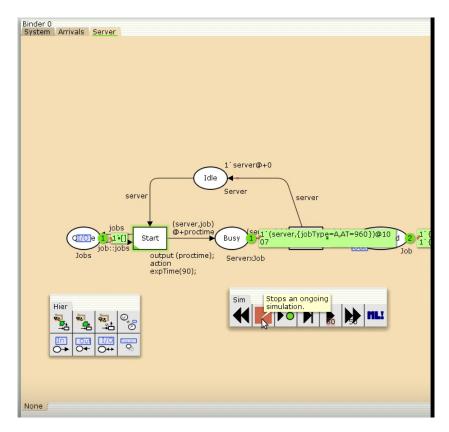


Рис. 3.5: Симуляция

Создаем 2 новых подраздела в Monitor - Ostanovka и Queue\_Delay (рис. 3.6)

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

Рис. 3.6: Ostanovka и Queue\_Delay

#### Запуск симуляции с новыми подразделами (рис. 3.7)

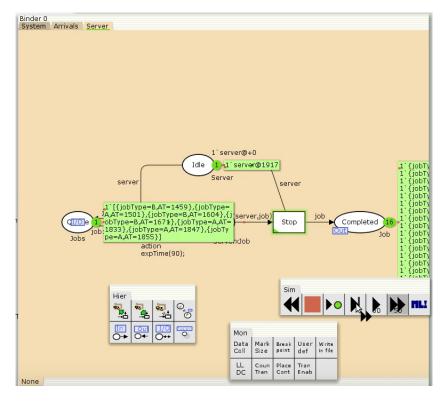


Рис. 3.7: Симуляция

Вывод с файла Queue\_Delay(рис. 3.8).

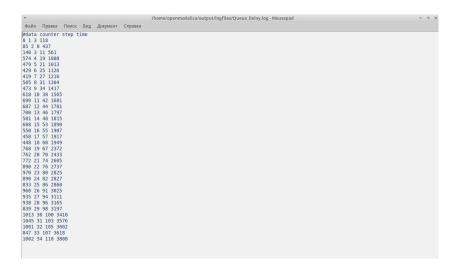


Рис. 3.8: Queue\_Delay

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)

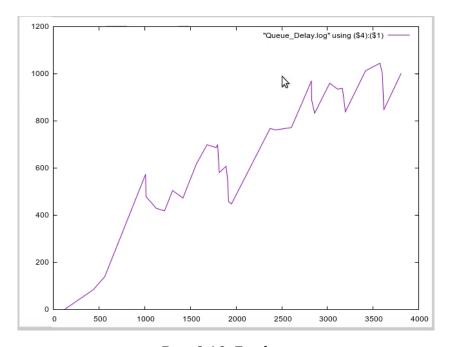


Рис. 3.10: График

Добавили новый подраздел Queue Delay Real (рис. 3.11)

```
▼ Queue Delay Real

► Type: Data collection

► Nodes ordered by pages

► Predicate

▼ Observer

fun obs (bindelem) =
let

fun obsBindElem (Server'Start (1, {job,jobs,proctime})) = Real.fromInt(intTime()-(#AT job))

| obsBindElem _ = ~1.0
in
obsBindElem bindelem
end

► Init function
```

Рис. 3.11: Queue Delay Real

Содержимое файла Queue Delay Real (рис. 3.12)

```
/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
```

Рис. 3.12: Queue Delay Real

Добавили новый подраздел Long Delay Time (рис. 3.13)

```
    ▼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. 3.13: Long Delay Time

Содержимое файла Long Delay Time (рис. 3.14)

```
/home/openmodelica/output/logfiles/Long_Delay_Time.log - Mousepad
 Файл Правка Поиск Вид Документ Справка
#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
```

Рис. 3.14: Long Delay Time

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

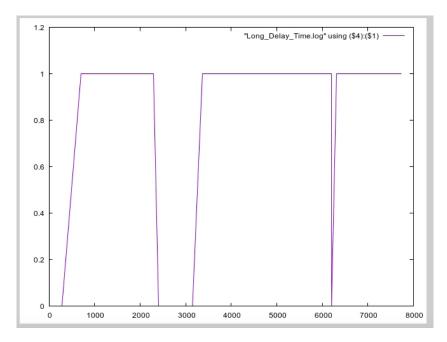


Рис. 3.15: График

# 4 Выводы

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