

Отчёт по лабораторной работе №11

Модель системы массового обслуживания $M|M|1$

Козлов Всеволод Павлович НФИбд-02-22

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 15 |

Список иллюстраций

| | | |
|------|--|----|
| 3.1 | Граф сети системы обработки заявки в очереди | 7 |
| 3.2 | Граф генератора заявок системы | 7 |
| 3.3 | Граф процесса обработки заявок на сервере системы | 8 |
| 3.4 | Декларация системы | 8 |
| 3.5 | Параметры элементов основного графика системы обработки заявок в очереди | 9 |
| 3.6 | Параметры элементов генератора заявок системы | 9 |
| 3.7 | Параметры элементов обработчика заявок системы | 9 |
| 3.8 | Функция Predicate монитора Ostanowka | 10 |
| 3.9 | Функция Observer монитора Queue Delay | 10 |
| 3.10 | Файл Queue_Delay.log | 11 |
| 3.11 | График изменения задержки в очереди | 11 |
| 3.12 | Функция Observer монитора Queue Delay Real | 12 |
| 3.13 | Файл Queue_Delay_Real.log | 12 |
| 3.14 | Функция Observer монитора Long Delay Time | 13 |
| 3.15 | Файл Long_Delay_Time.log | 13 |
| 3.16 | Периоды времени, когда значения задержки в очереди превышали заданное значение | 14 |

Список таблиц

1 Цель работы

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

2 Задание

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

3 Выполнение лабораторной работы

Граф сети системы обработки заявки в очереди (рис. 3.1)

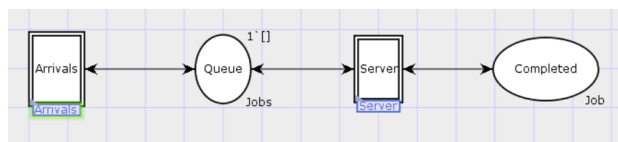


Рис. 3.1: Граф сети системы обработки заявки в очереди

Граф генератора заявок системы (рис. 3.2)

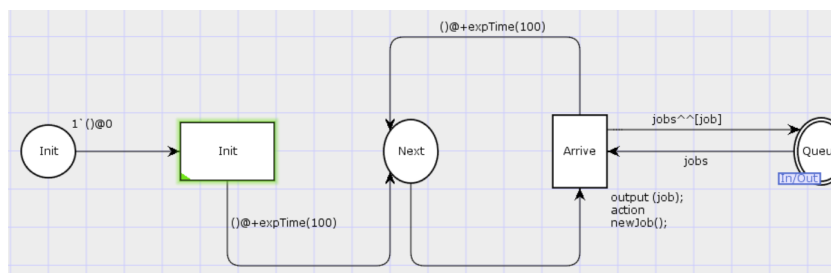


Рис. 3.2: Граф генератора заявок системы

Граф процесса обработки заявок на сервере системы (рис. 3.3)

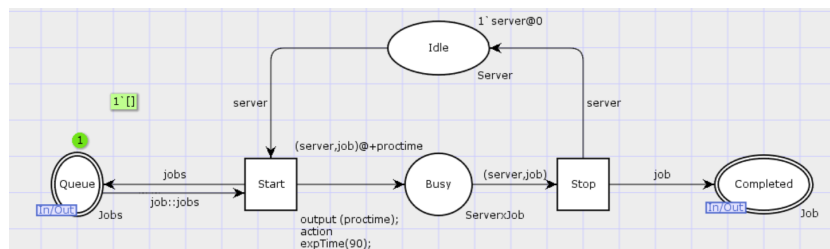


Рис. 3.3: Граф процесса обработки заявок на сервере системы

Задал декларацию системы(рис. 3.4)

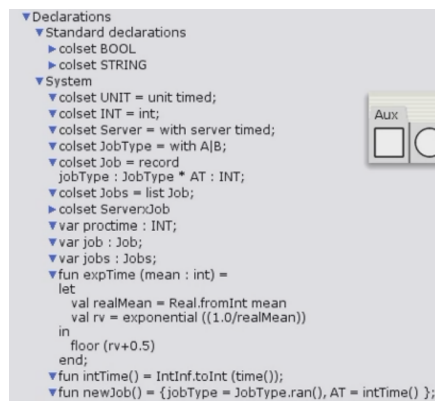


Рис. 3.4: Декларация системы

Параметры элементов основного графика системы обработки заявок в очереди
(рис. 3.5)

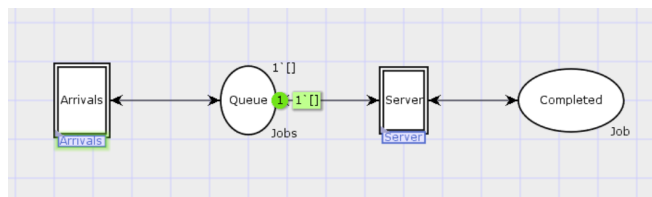


Рис. 3.5: Параметры элементов основного графика системы обработки заявок в очереди

Параметры элементов генератора заявок системы (рис. 3.6)

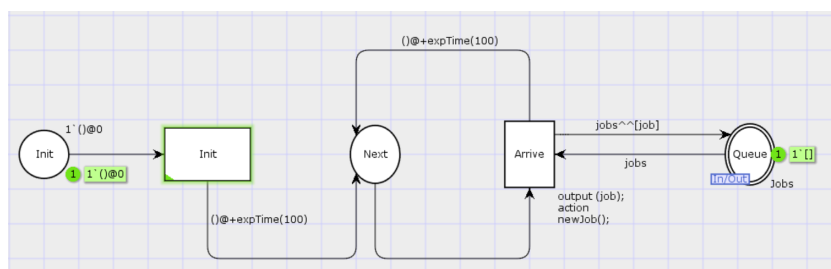


Рис. 3.6: Параметры элементов генератора заявок системы

Параметры элементов обработчика заявок системы (рис. 3.7)

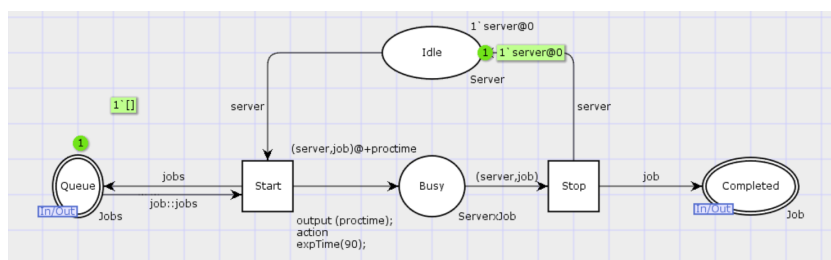
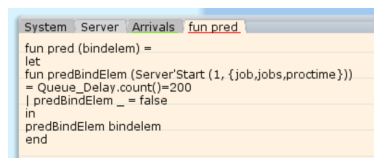


Рис. 3.7: Параметры элементов обработчика заявок системы

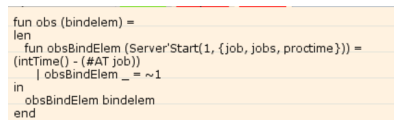
Написал функцию Predicate монитора Ostanowka (рис. 3.8)



```
System  Server  Arrivals  fun pred
fun pred (bindelem) =
let
  fun predBindElem (Server'Start(1, {job,jobs,proctime}))
    = Queue_Delay.count()=200
  | predBindElem _ = false
in
  predBindElem bindelem
end
```

Рис. 3.8: Функция Predicate монитора Ostanowka

Написал функцию Observer монитора Queue Delay (рис. 3.9)



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

Рис. 3.9: Функция Observer монитора Queue Delay

Получил следующий файл Queue_Delay.log (рис. 3.10)

```
#data counter step time
0 1 3 496
0 2 6 507
0 3 9 523
97 4 22 624
134 5 27 662
266 6 45 808
308 7 50 850
356 8 57 913
347 9 60 915
341 10 63 928
501 11 78 1085
664 12 97 1256
```

Рис. 3.10: Файл Queue_Delay.log

```
#!/usr/bin/gnuplot -persist
# задаём текстовую кодировку,
# тип терминала, тип и размер шрифта

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

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

Построил график изменения задержки в очереди (рис. 3.11)

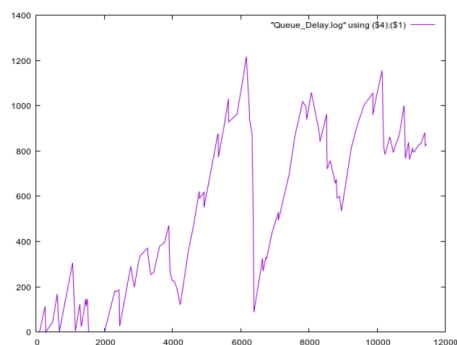


Рис. 3.11: График изменения задержки в очереди

Написал функцию Observer монитора Queue Delay Real (рис. 3.12)

```
fun obs (bindelem) =  
  len  
  fun obsBindElem (Server/Start(1, {job, jobs, proctime})) =  
    Real.fromInt(intTime()) - (#AT job)  
    | obsBindElem _ = ~1  
  in  
    obsBindElem bindelem  
  end
```

Рис. 3.12: Функция Observer монитора Queue Delay Real

Получил следующий файл Queue_Delay_Real.log (рис. 3.13)

```
#data counter step time  
0.000000 1 3 69  
60.000000 2 6 197  
135.000000 3 9 373  
75.000000 4 12 499  
0.000000 5 15 777  
12.000000 6 18 844  
0.000000 7 21 957  
175.000000 8 27 1167  
397.000000 9 30 1419  
459.000000 10 34 1537  
647.000000 11 39 1726  
536.000000 12 41 1879  
471.000000 13 43 1929  
690.000000 14 47 2201  
626.000000 15 49 2219  
626.000000 16 51 2231  
633.000000 17 54 2276  
358.000000 18 56 2289  
260.000000 19 58 2304  
650.000000 20 70 2924  
586.000000 21 72 2934  
709.000000 22 74 3059  
821.000000 23 76 3177
```

Рис. 3.13: Файл Queue_Delay_Real.log

Написал функцию Observer монитора Long Delay Time (рис. 3.14)

```

System  Server  Arrivals  fun pred  fun obs
fun obs (bindelem) =
if IntInf.toInt(Queue_Delay.Last())>=(llongdelaytime)
then 1
else 0

```

Рис. 3.14: Функция Observer монитора Long Delay Time

Получил следующий файл Long_Delay_Time.log (рис. 3.15)

```

#data counter step time
0 1 3 36
0 2 6 282
0 3 9 515
0 4 12 705
0 5 17 826
1 6 21 938
0 7 24 1003
0 8 26 1077
0 9 28 1107
0 10 31 1159
0 11 34 1285
0 12 37 1344
0 13 39 1435
0 14 42 1500
0 15 45 1530
0 16 48 1587
0 17 51 1644
0 18 55 1729
0 19 57 1769
0 20 60 1905
0 21 63 1944
0 22 66 2068
1 23 74 2396
1 24 76 2400
1 25 81 2528

```

Рис. 3.15: Файл Long_Delay_Time.log

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

```
# задаём текстовую кодировку,
```

```
# тип терминала, тип и размер шрифта
```

```
set encoding utf8
```

```
set term pngcairo font "Helvetica,9"
```

```
# задаём выходной файл графика
```

```
set out 'window_1.png'
```

```
set style line 2
```

```
plot [0:] [0:1.2] "Long_Delay_Time.log" using ($4):($1) with lines
```

Построил график: периоды времени, когда значения задержки в очереди превышали заданное значение (рис. 3.16)

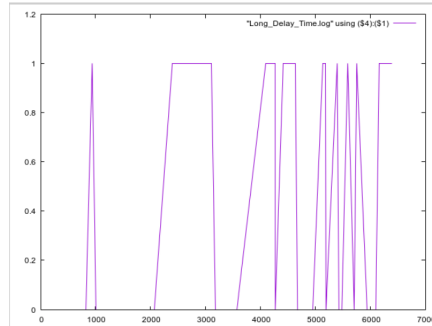


Рис. 3.16: Периоды времени, когда значения задержки в очереди превышали заданное значение

4 Выводы

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