

# Лабораторная работа 11

Модель системы массового обслуживания  $M|M|1$

---

Оразгелдиев Язгелди

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Оразгелдиев Язгелди
- студент
- Российский университет дружбы народов
- orazgeldiyev.yazgeldi@gmail.com
- <https://github.com/YazgeldiOrazgeldiyev>

Реализовать модель  $M|M|1$  в CPNTools

- Реализовать модель системы массового обслуживания  $M|M|1$
- Настроить мониторинг параметров моделируемой системы и нарисовать графики очереди

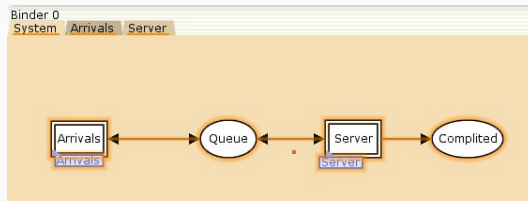


Рис. 1: Граф сети системы обработки заявок в очереди

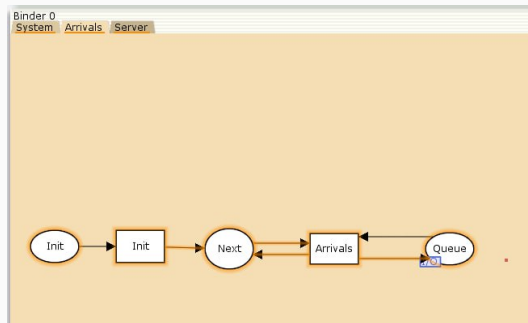


Рис. 2: Граф генератора заявок системы

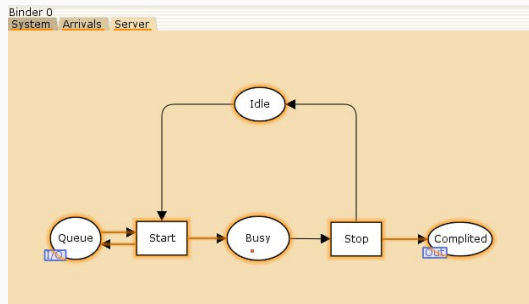


Рис. 3: Граф процесса обработки заявок на сервере системы



```
▶ Tool box
▶ Help
▶ Options
▼ cont111.con
  Step: 0
  Time: 0
  ▶ Options
  ▶ History
  ▼ Declarations
    ▼ Standard dedarations
      ▶ colset STRING
      ▶ colset BOOL
    ▼ System
      ▶ colset UNIT
      ▼ 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 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. 4: Задание декларации системы

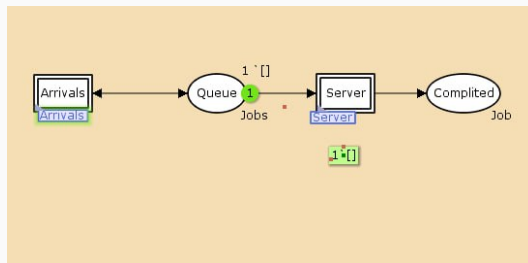


Рис. 5: Параметры элементов основного графа системы обработки заявок в очереди

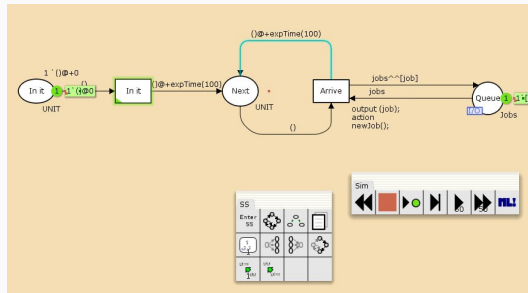


Рис. 6: Параметры элементов генератора заявок системы

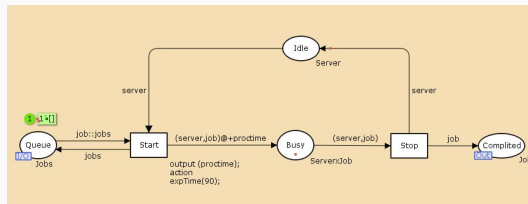


Рис. 7: Параметры элементов обработчика заявок системы

```
Binder 0
System Arrivals Server fun pred<Ostanovka> fun expTime colset Job fun obs
fun pred (bindelem) =
  let
    fun predBindElem (Server'Start (1,
                                   {job,jobs,proctime})) = Queue_Delay.count()=200
    | predBindElem _ = false
  in
    predBindElem bindelem
  end
```

Рис. 8: Функция Predicate монитора Остановка

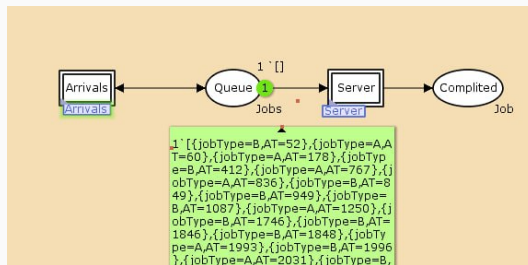


Рис. 9: Запуск системы обработки заявок в очереди

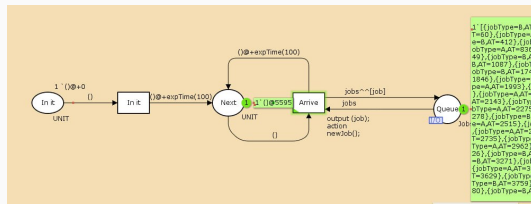


Рис. 10: Запуск системы обработки заявок в очереди

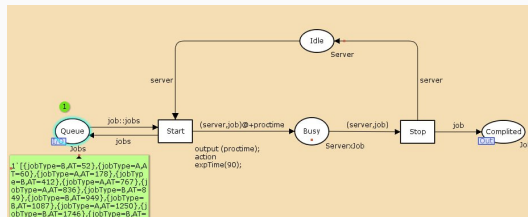
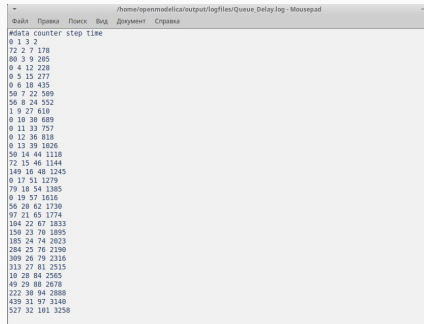


Рис. 11: Запуск системы обработки заявок в очереди



```
Binder 0
fun obs
  fun obs (bindelem) =
  let
    fun obsBindElem (Server'Start (1, {job,jobs,proctime})) =
      (intTime() - (#AT job))
      | obsBindElem _ = ~1
    in
      obsBindElem bindelem
    end
```

Рис. 12: Функция Observer монитора Queue Delay



```

#data counter step time
0 1 3 2
72 2 7 178
80 3 9 205
0 4 12 228
0 5 15 277
0 6 18 435
50 7 22 509
56 8 24 552
1 9 27 610
0 10 30 689
0 11 33 757
0 12 36 818
0 13 39 1026
50 14 44 1118
72 15 46 1144
149 16 48 1245
0 17 51 1279
79 18 54 1385
0 19 57 1616
56 20 62 1730
97 21 65 1774
104 22 67 1833
150 23 70 1895
185 24 74 2023
284 25 76 2190
309 26 79 2316
313 27 81 2515
10 28 84 2565
49 29 88 2678
222 30 94 2888
439 31 97 3140
527 32 101 3258

```

Рис. 13: Файл Queue\_Delay.log

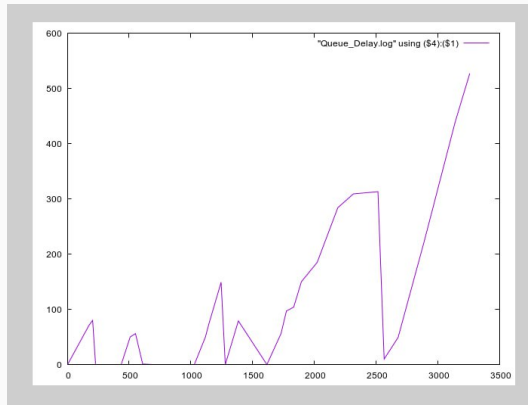
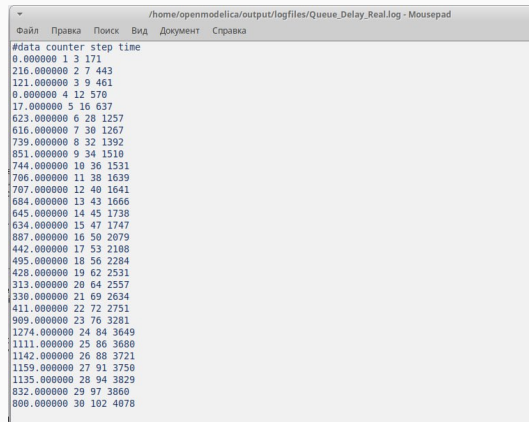


Рис. 14: График изменения задержки в очереди

```
server 1
fun obs <Queue Delay Real>
fun obs (bindelem) =
let
  fun obsBindElem (Server'Start (1, {job,jobs,proctime})) =
    Real.fromInt(intTime()-(#AT job))
    | obsBindElem _ = ~1.0
in
  obsBindElem bindelem
end
```

Рис. 15: Функция Observer монитора Queue Delay Real



```
#data counter step time
0.000000 1 3 171
216.000000 2 7 443
121.000000 3 9 461
0.000000 4 12 570
17.000000 5 16 637
623.000000 6 28 1257
616.000000 7 30 1267
739.000000 8 32 1392
851.000000 9 34 1510
744.000000 10 36 1531
706.000000 11 38 1639
707.000000 12 40 1641
684.000000 13 43 1666
645.000000 14 45 1738
634.000000 15 47 1747
887.000000 16 50 2079
442.000000 17 53 2108
495.000000 18 56 2284
428.000000 19 62 2531
313.000000 20 64 2557
330.000000 21 69 2634
411.000000 22 72 2751
909.000000 23 76 3281
1274.000000 24 84 3649
1111.000000 25 86 3680
1142.000000 26 88 3721
1159.000000 27 91 3750
1135.000000 28 94 3829
832.000000 29 97 3860
800.000000 30 102 4078
```

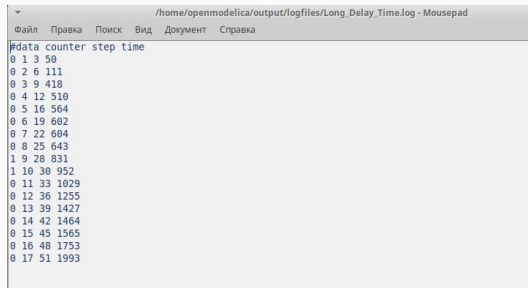
Рис. 16: Файл Queue\_Delay\_Real.log

```
fun obs (bindelem) =  
  if IntInf.toInt(Queue_Delay.last()) >= (!longdelaytime)  
  then 1  
  else 0
```

Рис. 17: Функция Observer монитора Long Delay Time

- ▼ Declarations
  - ▼ Standard declarations
    - ▶ colset STRING
    - ▶ colset BOOL
  - ▶ System
  - ▼ globref longdelaytime = 200;
- ▼ Monitors

Рис. 18: Функция Observer монитора Long Delay Time



```
/home/openmodelica/output/logfiles/Long_Delay_Time.log - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
#data counter step time
0 1 3 50
0 2 6 111
0 3 9 418
0 4 12 510
0 5 16 564
0 6 19 602
0 7 22 604
0 8 25 643
1 9 28 831
1 10 30 952
0 11 33 1029
0 12 36 1255
0 13 39 1427
0 14 42 1464
0 15 45 1565
0 16 48 1753
0 17 51 1993
```

Рис. 19: Файл Long\_Delay\_Time.log



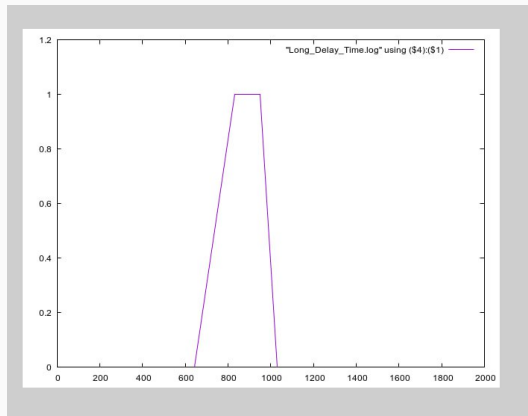


Рис. 20: Периоды времени, когда значения задержки в очереди превышали заданное значение

- В процессе лабораторной работы мы реализовали модель системы массового обслуживания  $M|M|1$  в CPNTools