

# Лабораторная работа №11

Модель системы массового обслуживания  $M|M|1$

---

Алади П. Ч.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Алади Принц Чисом
  - студент
  - Российский университет дружбы народов
  - 103225007@pfur.ru
  - <https://pjosh456.github.io/>

## Вводная часть

---

### Цель работы

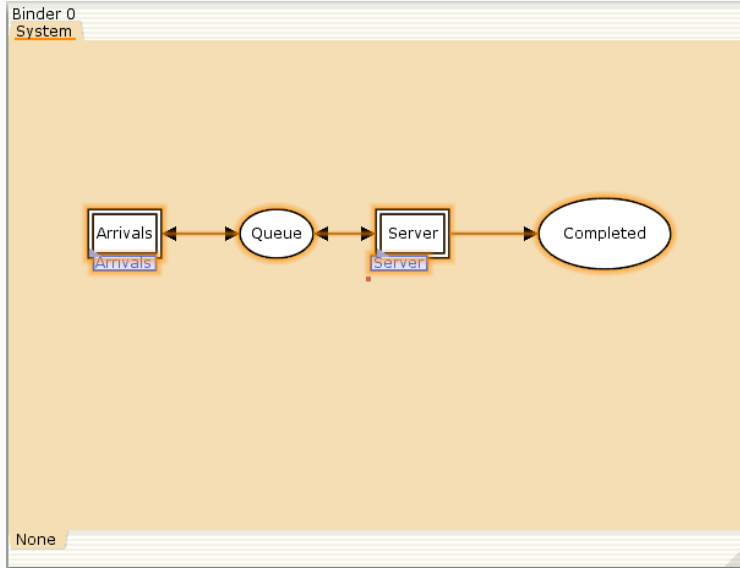
Реализовать модель  $M|M|1$  в CPN tools.

### Задание

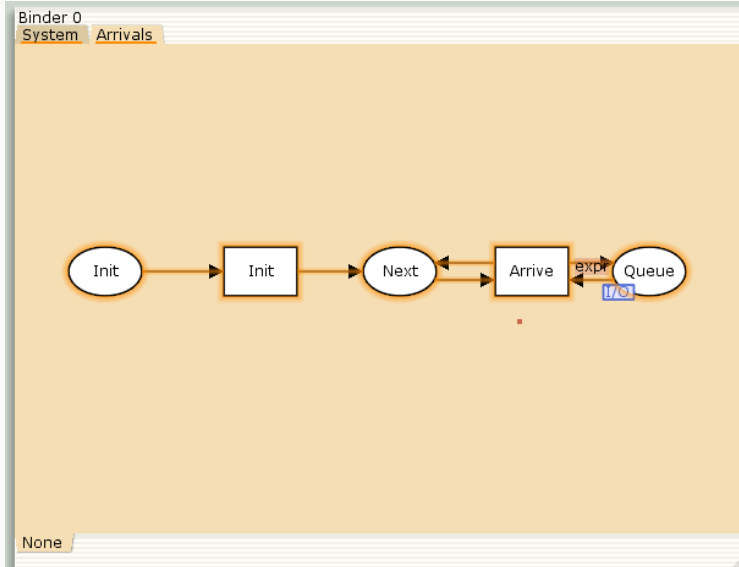
- Реализовать в CPN Tools модель системы массового обслуживания  $M|M|1$ .
- Настроить мониторинг параметров моделируемой системы и нарисовать графики очереди.

### Постановка задачи

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.

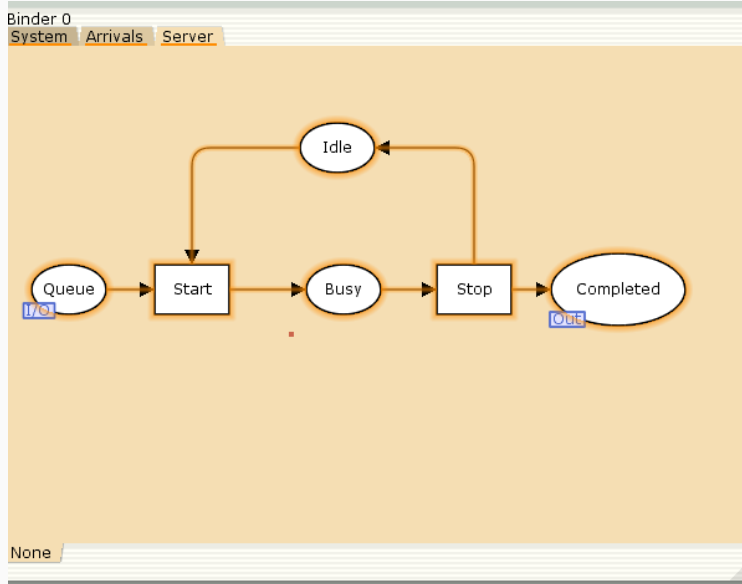


## Выполнение лабораторной работы





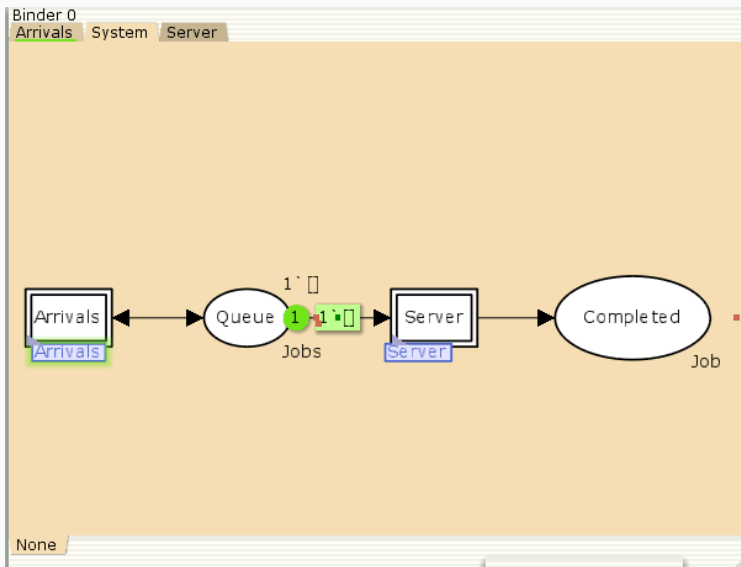
## Выполнение лабораторной работы



```
▼ Declarations
  ▼ Standard declarations
    ► colset BOOL
    ▼ colset STRING = string;
  ▼ System
    ▼ colset UNIT = unit timed;
    ▼ 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 ServersJob = 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()};
```

Рис. 4: Задание деклараций системы

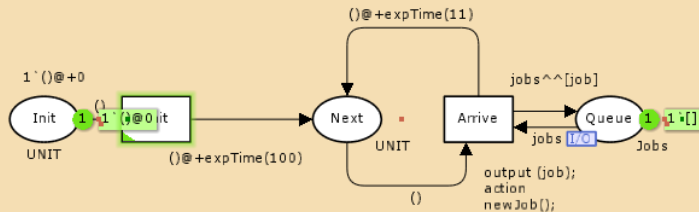
## Выполнение лабораторной работы



# Выполнение лабораторной работы

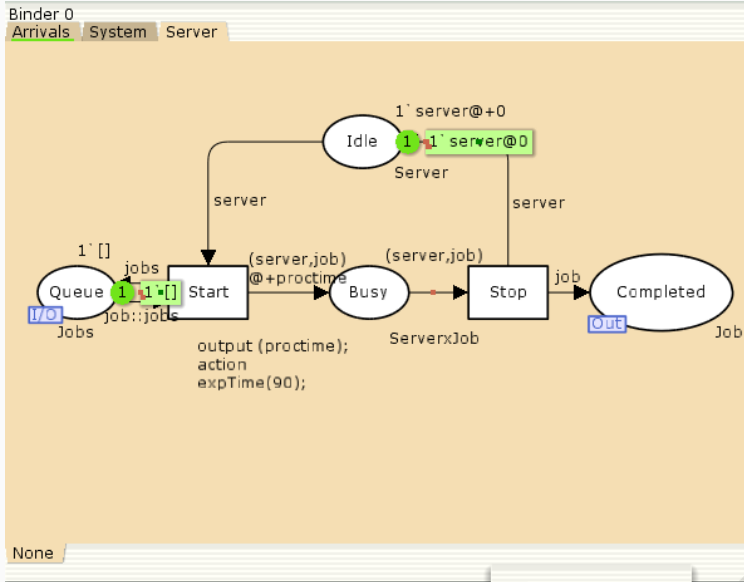
Binder 0

Arrivals System Server



None

## Выполнение лабораторной работы



Binder 0

Arrivals System Server fun pred <Ostanovka>

```
fun pred (bindelem) =
```

```
let
```

```
  fun predBindElem (Server'Start (1,  
                                   {job,jobs,proctime})) = Queue_Delay.count()=200|  
    | predBindElem _ = false
```

```
in
```

```
  predBindElem bindelem
```

```
end
```

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

Рис. 9: Функция Observer монитора Queue Delay

```
Файл  Правка  Поиск  Вид  Документ  Справка
1 #data counter step time
2 0 1 3 106
3 17 2 6 129
4 0 3 9 186
5 58 4 15 249
6 74 5 19 273
7 179 6 26 384
8 166 7 28 405
9 165 8 32 423
10 948 9 111 1209
11 1169 10 116 1444
12 |
```

Рис. 10: Файл Queue\_Delay.log



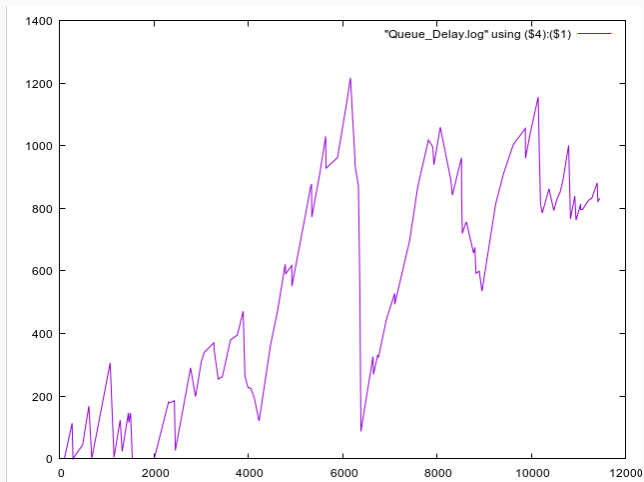
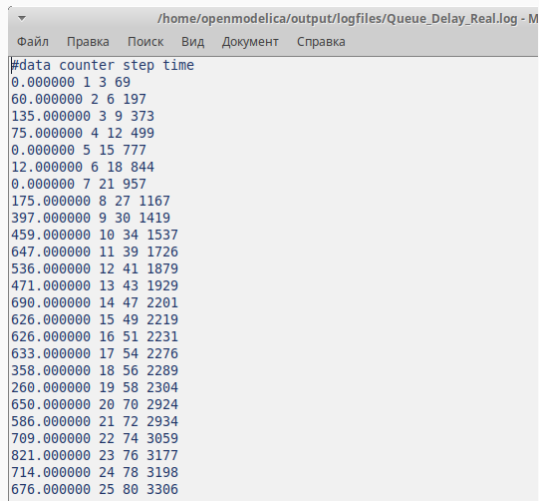


Рис. 11: График изменения задержки в очереди

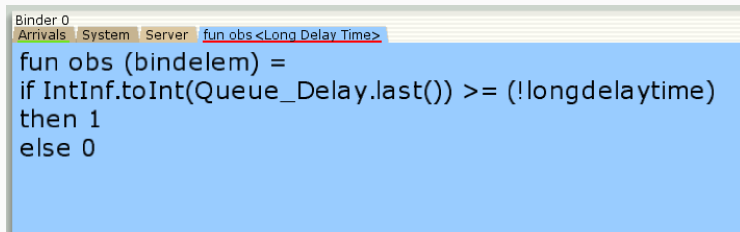
```
Binder 0  
Arrivals System Server fun obs <Queue Delay Real>  
fun obs (bindelem) =  
  let  
    fun obsBindElem (Server'Start (1, {job,jobs,proctime})) =  
      Real.fromInt(intTime()-(#AT job))  
      | obsBindElem _ = ~1.0  
  in  
    obsBindElem bindelem  
  end
```

Рис. 12: Функция Observer монитора Queue Delay Real



```
/home/openmodelica/output/logfiles/Queue_Delay_Real.log - M
Файл  Правка  Поиск  Вид  Документ  Справка
#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
714.000000 24 78 3198
676.000000 25 80 3306
```

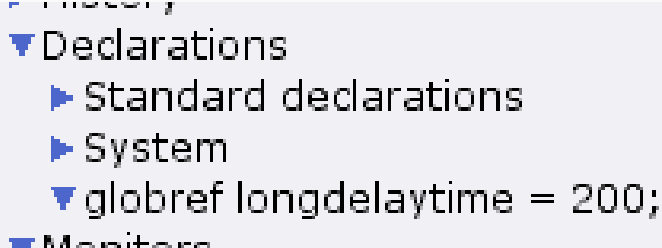
Рис. 13: Содержимое Queue\_Delay\_Real.log



The image shows a screenshot of a code editor window. At the top, there is a tab bar with four tabs: 'Arrivals' (highlighted in green), 'System' (highlighted in orange), 'Server' (highlighted in yellow), and 'fun obs <Long Delay Time>' (highlighted in blue). Below the tabs, the code is written in a light blue background. The code defines a function 'fun obs (bindelem) =' which checks if the last element of 'Queue\_Delay' is greater than or equal to 'longdelaytime'. If true, it returns 1; otherwise, it returns 0.

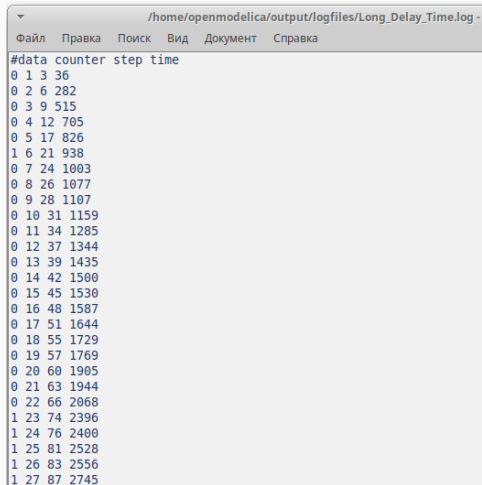
```
Binder 0  
Arrivals System Server fun obs <Long Delay Time>  
fun obs (bindelem) =  
if IntInf.toInt(Queue_Delay.last()) >= (!longdelaytime)  
then 1  
else 0
```

Рис. 14: Функция Observer монитора Long Delay Time



```
▼ Declarations
  ► Standard declarations
  ► System
  ▼ globref longdelaytime = 200;
▼ Monitors
```

Рис. 15: Определение longdelaytime в декларациях



```

/home/openmodelica/output/logfiles/Long_Delay_Time.log -
Файл  Правка  Поиск  Вид  Документ  Справка
#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
1 26 83 2556
1 27 87 2745

```

Рис. 16: Содержимое Long\_Delay\_Time.log

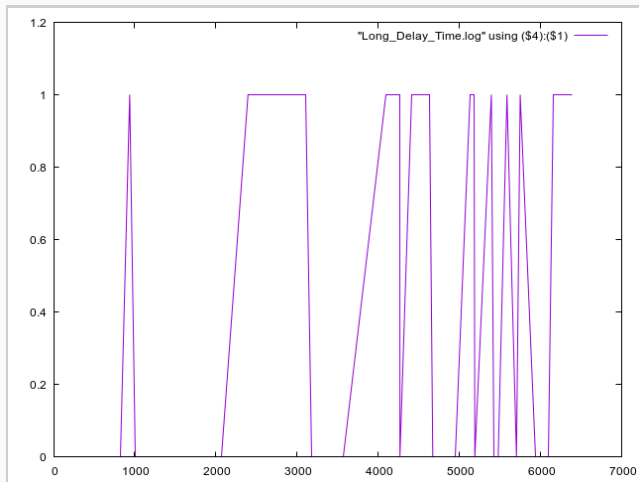


Рис. 17: Периоды времени, когда значения задержки в очереди превышали заданное значение

В процессе выполнения данной лабораторной работы я реализовала модель системы массового обслуживания  $M|M|1$  в CPN Tools.