

Лабораторная работа № 11

Модель системы массового обслуживания $M|M|1$

Шияпова Д.И.

05 апреля 2025

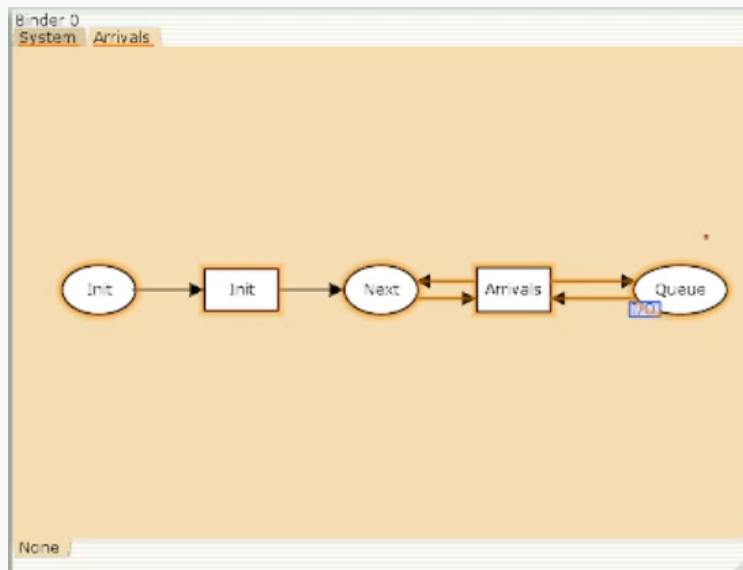
Российский университет дружбы народов, Москва, Россия

- Шияпова Дарина Илдаровна
- Студентка
- Российский университет дружбы народов
- 1132226458@pfur.ru

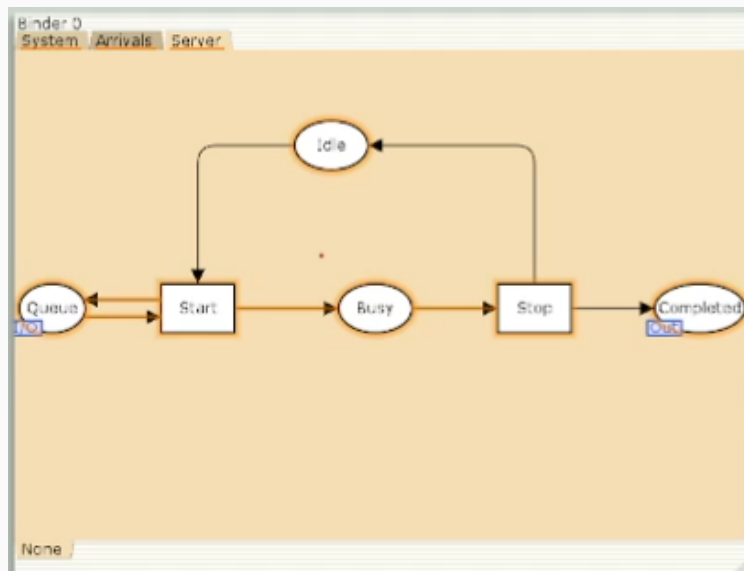


Реализовать модель $M|M|1$ в CPN tools.

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



Зададим декларации системы.

Определим множества цветов системы (colorset):

- фишки типа **UNIT** определяют моменты времени;
- фишки типа **INT** определяют моменты поступления заявок в систему.
- фишки типа **JobType** определяют 2 типа заявок — A и B;
- кортеж **Job** имеет 2 поля: jobType определяет тип работы (соответственно имеет тип **JobType**, поле AT имеет тип **INT** и используется для хранения времени нахождения заявки в системе);
- фишки **Jobs** — список заявок;
- фишки типа **ServerxJob** — определяют состояние сервера, занятого обработкой заявок.

Переменные модели:

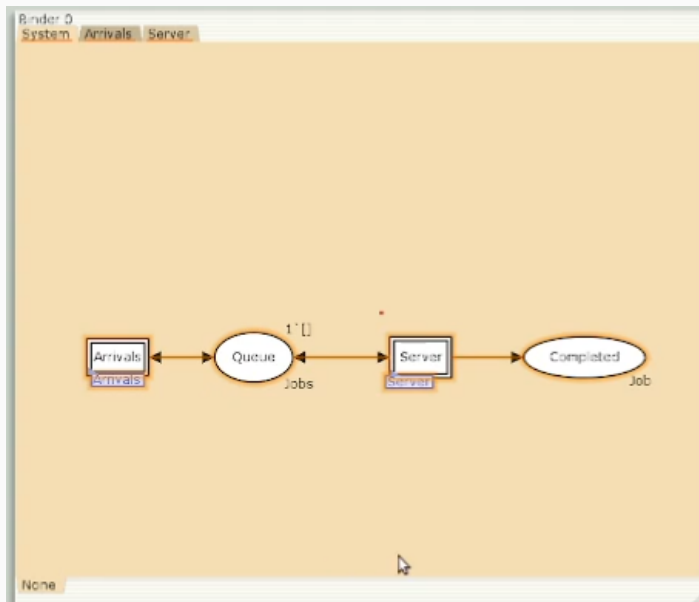
- `proctime` — определяет время обработки заявки;
- `job` — определяет тип заявки;
- `jobs` — определяет поступление заявок в очередь.

Определим функции системы:

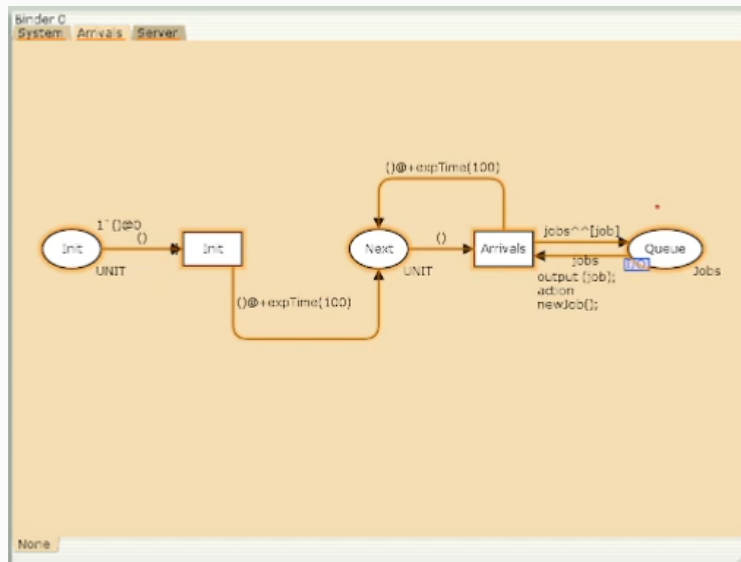
- функция **expTime** описывает генерацию целочисленных значений через интервалы времени, распределённые по экспоненциальному закону;
- функция **intTime** преобразует текущее модельное время в целое число;
- функция **newJob** возвращает значение из набора **Job** — случайный выбор типа заявки (А или В).

```
▼ Declarations
  ► SYSTEM
  ► globref longdelaytime = 200;
  ► colset UNIT
  ► colset INT
  ▼ colset Server = with server timed;
  ► colset JobType
  ▼ colset Job = record jobType : JobType *
    AT : INT;
  ▼ colset Jobs = list Job;
  ► colset ServerxJob
  ▼ var proctime : INT;
  ► var job
  ► var jobs
  ► fun expTime
  ► fun intTime
  ► fun newJob
  ► Standard declarations
```

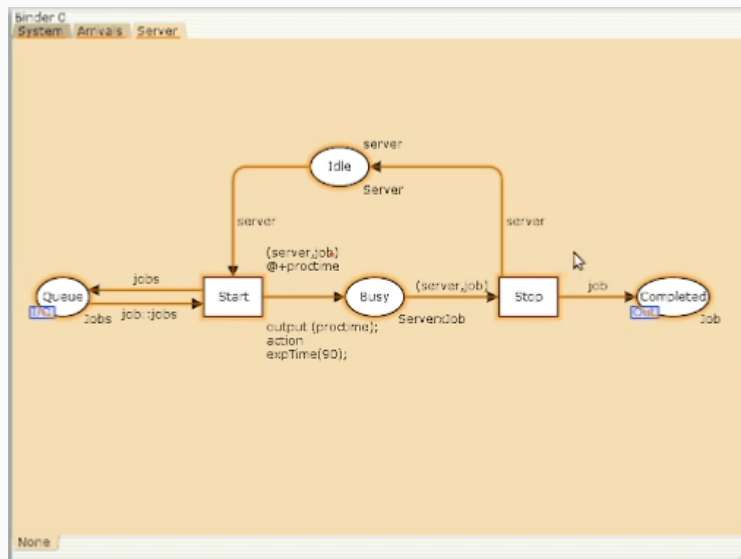
Выполнение лабораторной работы

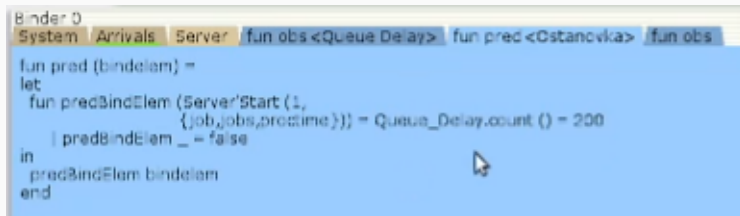


Выполнение лабораторной работы



Выполнение лабораторной работы

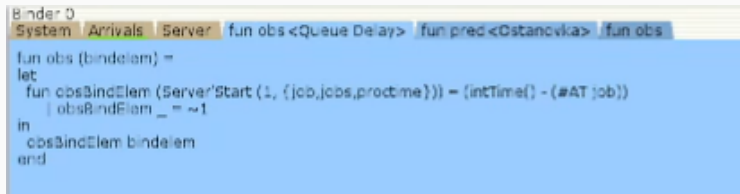




The screenshot shows a code editor window titled "Binder 0". It has several tabs: "System", "Arrivals", "Server", "fun obs <Queue Delay>", "fun pred <Ostanovka>", and "fun obs". The "fun pred <Ostanovka>" tab is selected, displaying the following code:

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

Рис. 7: Функция Predicate монитора Ostanovka



The screenshot shows a NetLogo code editor with a tab labeled 'Binder 0'. Below the tab, there are several buttons: 'System', 'Arrivals', 'Server', 'fun obs <Queue Delay>', 'fun pred <Ostankovka>', and 'fun obs'. The 'fun obs <Queue Delay>' button is selected. The code area below the buttons contains the following text:

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

Рис. 8: Функция Observer монитора Queue Delay

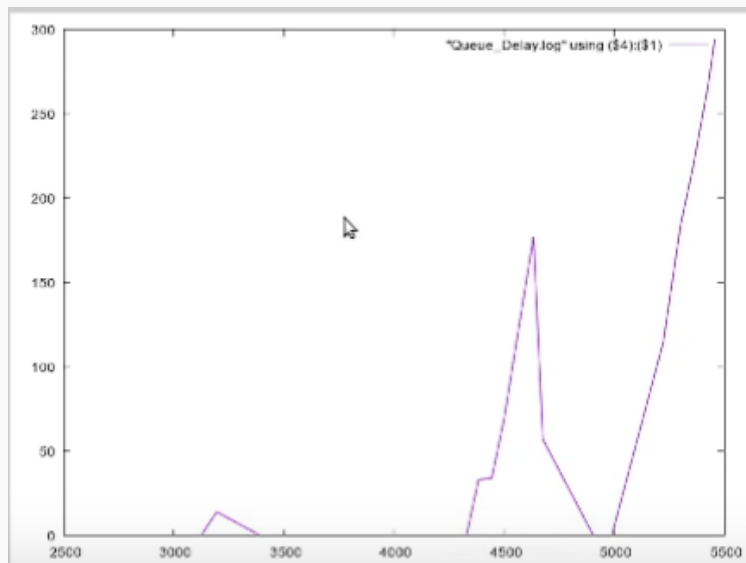
```
data counter step time
0 23 69 2510
0 24 72 2637
0 25 75 2807
0 26 78 3123
14 27 81 3196
0 28 84 3390
0 29 87 3945
0 30 90 4327
13 31 93 4383
14 32 97 4443
69 33 100 4499
```

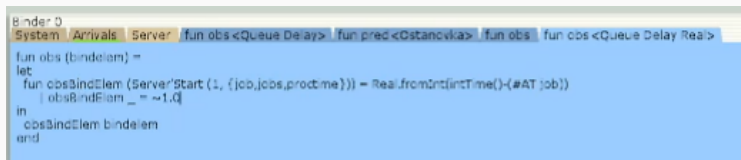
Рис. 9: Файл Queue_Delay.log


```
#!/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
```





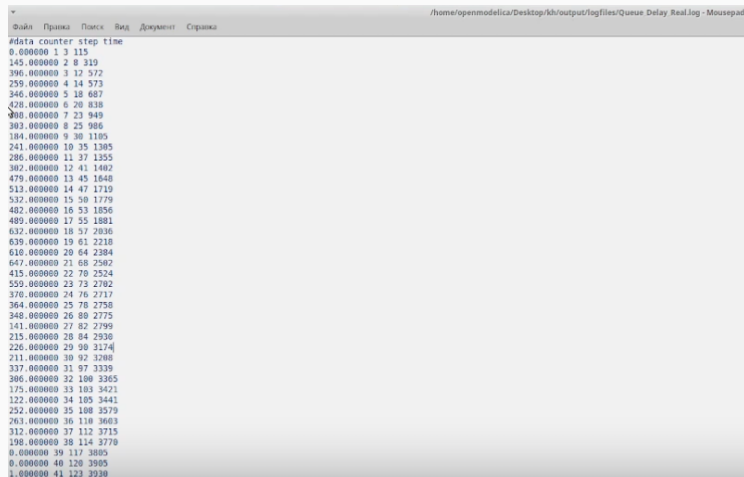
The screenshot shows a NetLogo environment with a code editor. The editor has a tab labeled 'fun obs <Queue Delay>'. The code in the editor is as follows:

```
Binder 0
System Arrivals Server fun obs <Queue Delay> fun pred <Ostancivka> fun obs 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
```

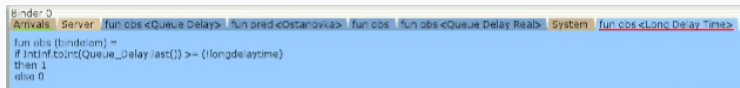
Рис. 11: Функция Observer монитора Queue Delay Real

Выполнение лабораторной работы



```
#data counter step time
0.000000 1 3 115
145.000000 2 8 319
396.000000 3 12 572
259.000000 4 14 573
346.000000 5 18 687
428.000000 6 20 838
500.000000 7 23 949
603.000000 8 25 986
184.000000 9 30 1105
241.000000 10 35 1305
286.000000 11 37 1355
302.000000 12 41 1402
479.000000 13 45 1648
513.000000 14 47 1719
532.000000 15 50 1779
482.000000 16 53 1856
489.000000 17 55 1881
632.000000 18 57 2036
639.000000 19 61 2218
610.000000 20 64 2384
647.000000 21 68 2502
415.000000 22 70 2524
559.000000 23 73 2702
370.000000 24 76 2717
364.000000 25 78 2758
348.000000 26 80 2775
141.000000 27 82 2799
215.000000 28 84 2930
226.000000 29 90 3174
211.000000 30 92 3208
337.000000 31 97 3339
306.000000 32 100 3365
175.000000 33 103 3421
122.000000 34 105 3441
252.000000 35 108 3579
263.000000 36 110 3603
312.000000 37 112 3715
198.000000 38 114 3770
0.000000 39 117 3885
0.000000 40 120 3905
1.000000 41 123 3930
```

Рис. 12: Содержимое Queue_Delay_Real.log



The screenshot shows a NetLogo environment with a command window. The command window has a tab labeled 'fun obs <Long Delay Time>' which is highlighted in red. The code in the command window is as follows:

```
Binder 0  
Arrivals Server fun obs <Queue Delay> fun pred <Ostanovka> fun obs fun obs <Queue Delay Reab> System fun obs <Long Delay Time>  
fun obs (bindelam) =  
if intinf.toInt(Queue_Delay/ast()) >= (longdelaytime)  
then 1  
else 0
```

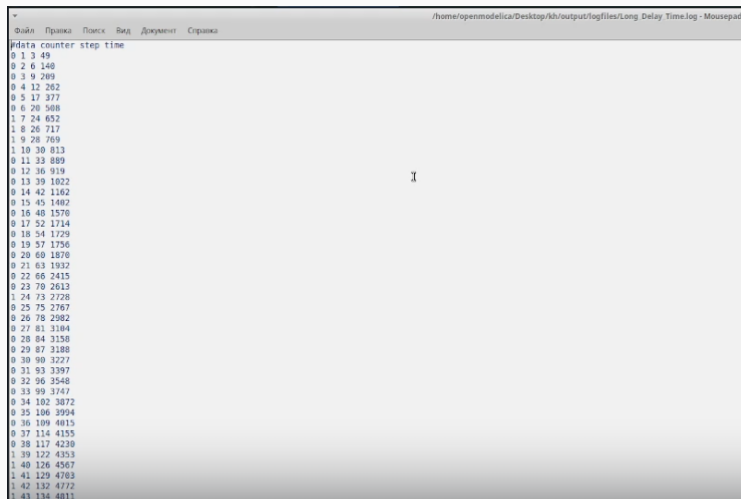
Рис. 13: Функция Observer монитора Long Delay Time

▼ Declarations

- ▶ SYSTEM
- ▶ global longdelaytime = 200;
- ▶ colset UNIT

Рис. 14: Определение longdelaytime в декларациях

Выполнение лабораторной работы



```
Файл  Правка  Поиск  Вид  Документ  Справка
/home/openmodelica/Desktop/kh/output/logfiles/Long_Delay_Time.log - Mousepad
data counter step time
0 1 3 49
0 2 6 140
0 3 9 209
0 4 12 262
0 5 17 377
0 6 20 508
1 7 24 652
1 8 26 717
1 9 28 769
1 10 30 813
0 11 33 889
0 12 36 919
0 13 39 1022
0 14 42 1162
0 15 45 1402
0 16 48 1570
0 17 52 1714
0 18 54 1729
0 19 57 1756
0 20 60 1870
0 21 63 1932
0 22 66 2415
0 23 70 2613
1 24 73 2728
0 25 75 2767
0 26 78 2982
0 27 81 3104
0 28 84 3150
0 29 87 3188
0 30 90 3227
0 31 93 3397
0 32 96 3548
0 33 99 3747
0 34 102 3872
0 35 106 3994
0 36 109 4015
0 37 114 4155
0 38 117 4230
1 39 122 4353
1 40 126 4567
1 41 129 4703
1 42 132 4772
1 43 134 4811
```

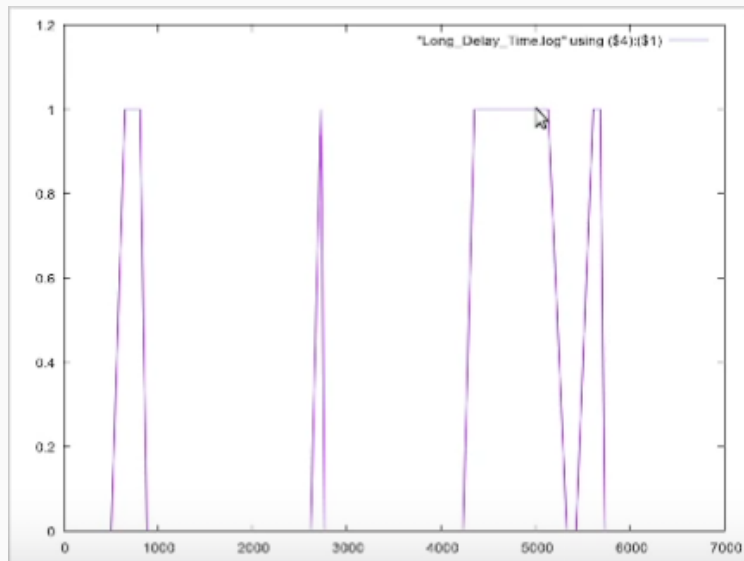
Рис. 15: Содержимое Long_Delay_Time.log

```
#!/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
```


Выполнение лабораторной работы



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