

# Презентация по лабораторной работе 11

Модель системы массового обслуживания  $M|M|1$

---

Сидорова Н.А.

19 апреля 2025

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

Сеть Петри для самой системы, сервер и генератор заявок описываются отдельными сетями, также здесь есть позиции очередь и обслуженная заявка

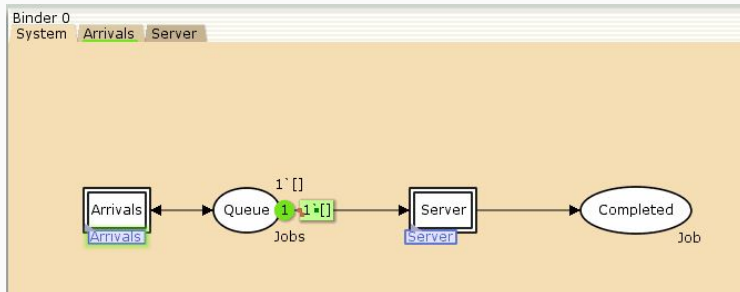


Рис. 1: Общая система

Сеть Петри для генератора заявок. Имеет позиции текущая заявка, следующая заявка и очередь. Два перехода: распределение поступления заявок и определение поступления заявки в очередь

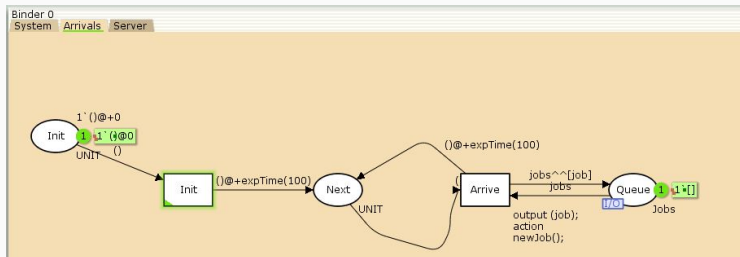


Рис. 2: Генератор заявок

Сеть Петри для сервера. Имеет позиции: очередь, сервер занят, сервер простаивает и заявка выполнена. Переходы: старт и стоп работы сервера.

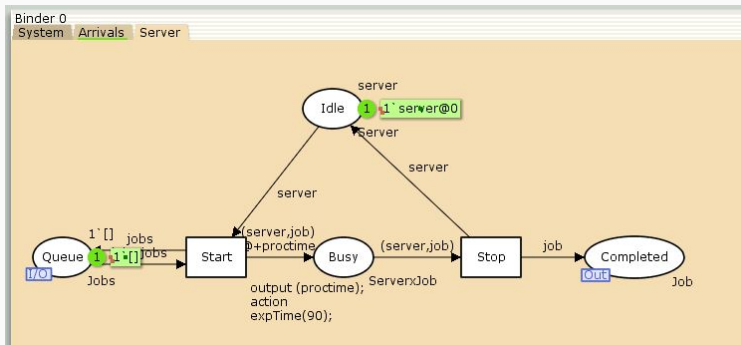


Рис. 3: Сервер

Декларации: определения множеств, инициализация переменных, объявление функций

```
Options
L11.cpn
  Step: 0
  Time: 0
Options
History
Declarations
  Standard declarations
    colset BOOL
    colset STRING
  System
    colset UNIT
    colset INT = int;
    colset Server = with server timed;
    colset JobType = with A | B;
    colset Job = record jobType : JobType * AT : INT;
    colset ServerxJob
    colset Jobs = list Job;
    var proctime : INT;
    var jobs : Jobs;
    var job : Job;
    fun intTime() = IntInf.toInt (time());
    fun newJob() = {jobType = JobType.ran(), AT = intTime()};
    fun expTime (mean: int) =
      let
        val realMean = Real.fromInt mean
        val rv = exponential((1.0/realMean))
      in
        floor (rv+0.5)
      end;
```

## Моделирование системы

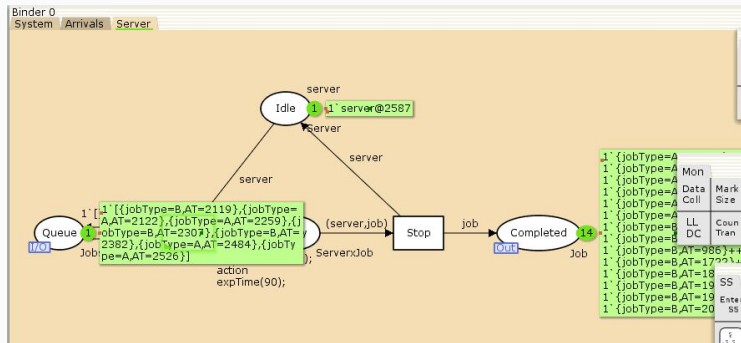


Рис. 5: Моделирование

Мониторинг параметров очереди. Изменение функции Observer чтобы получить значение задержки в очереди

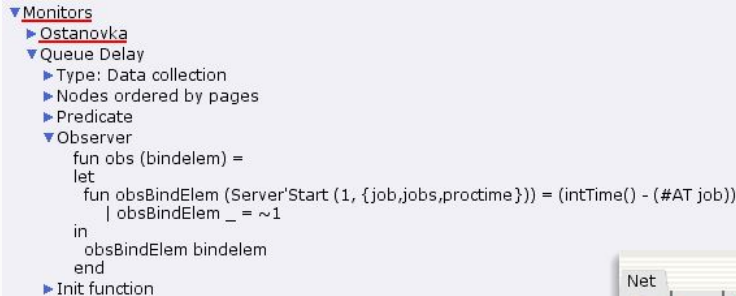
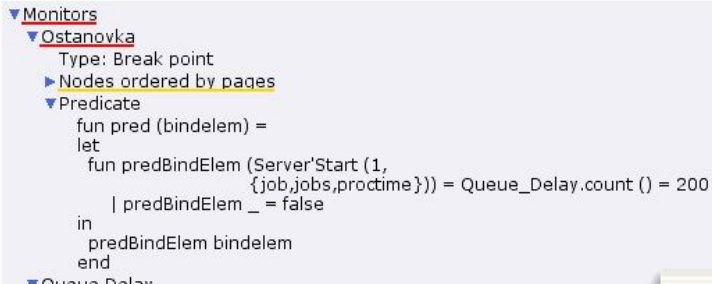


Рис. 6: Observer

Изменение функции Predicate чтобы останавливать мониторинг



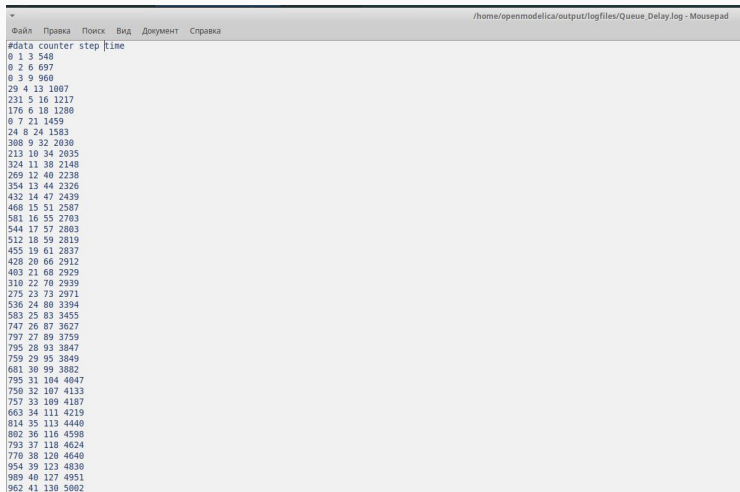
```
▼ Monitors  
  ▼ Ostanovka  
    Type: Break point  
    ► Nodes ordered by pages  
    ▼ Predicate  
      fun pred (bindelem) =  
      let  
        fun predBindElem (Server'Start (1,  
                                   {job,jobs,proctime})) = Queue_Delay.count () = 200  
          | predBindElem _ = false  
      in  
        predBindElem bindelem  
      end  
    ▼ Queue Delay
```

Рис. 7: Predicate



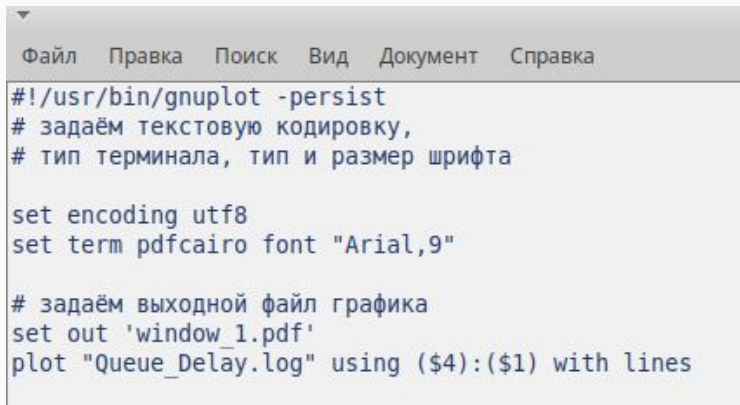
# Queue Delay.log

Значения файла Queue Delay.log, содержащие значение задержки очереди, счетчик, шаг, время



#data	counter	step	time
0	1	3	548
0	2	6	697
0	3	9	960
29	4	13	1007
231	5	16	1217
176	6	18	1280
0	7	21	1459
24	8	24	1583
308	9	32	2030
213	10	34	2035
324	11	38	2148
269	12	40	2238
354	13	44	2326
432	14	47	2439
468	15	51	2587
581	16	55	2703
544	17	57	2803
512	18	59	2819
455	19	61	2837
428	20	66	2912
403	21	68	2929
310	22	70	2939
275	23	73	2971
536	24	80	3394
583	25	83	3455
747	26	87	3627
797	27	89	3759
795	28	93	3847
759	29	95	3849
681	30	99	3882
795	31	104	4047
750	32	107	4133
757	33	109	4187
663	34	111	4219
814	35	113	4440
802	36	116	4598
793	37	118	4624
770	38	120	4640
954	39	123	4830
989	40	127	4951
962	41	130	5002

Код для создания графика задержки в очереди

A screenshot of a gnuplot terminal window. The window has a menu bar with the following items: 'Файл', 'Правка', 'Поиск', 'Вид', 'Документ', and 'Справка'. Below the menu bar, the following code is entered:

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

set encoding utf8
set term pdfcairo font "Arial,9"

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

Рис. 9: Код графика

## График задержки в очереди

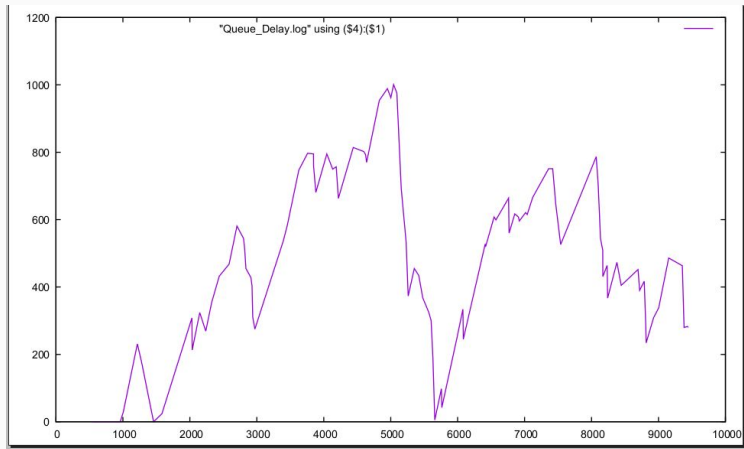


Рис. 10: График

Добавляем монитор Queue Delay Real и изменяем функцию Observer для действительных значений



```
▼ Monitors
  ▼ 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. 11: Queue Delay Real

Значения Queue Delay Real.log

Файл	Правка	Поиск	Вид	Документ	Справка
#data counter step time					
0.000000	1	3	63		
0.000000	2	6	248		
0.000000	3	9	287		
0.000000	4	12	792		
0.000000	5	15	1046		
76.000000	6	18	1126		
0.000000	7	21	1344		
98.000000	8	27	1461		
161.000000	9	29	1536		
182.000000	10	32	1597		
264.000000	11	35	1700		
149.000000	12	37	1718		
129.000000	13	39	1776		
26.000000	14	43	1821		
210.000000	15	46	2031		

## Long Delay Time

Добавляем монитор Long Delay Time и изменяем функцию Observer чтобы посчитать сколько раз задержка превысила заданное значение

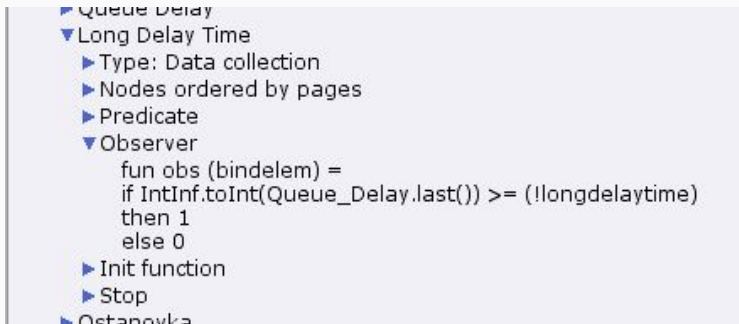


Рис. 13: Long Delay Time

Добавляем новую переменную в декларации

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

Рис. 14: Новая переменная

Код для графика в какие периоды значения задержки в очереди превышали заданное

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

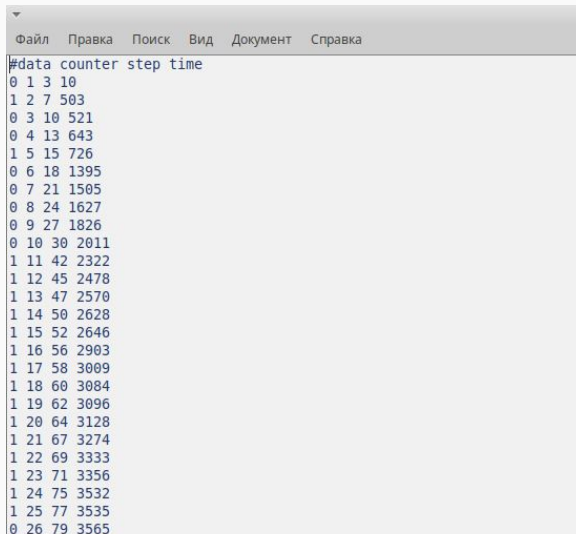
set encoding utf8
set term pdfcairo font "Arial,9"

# задаём выходной файл графика
set out 'window 2.pdf'
plot [0:][0:1.2] "Long_Delay_Time.log" using ($4):($1) with lines
```

Рис. 15: Код



## Значения Long Delay Time log



The image shows a screenshot of a text editor window. The title bar is grey and contains the text 'Файл Правка Поиск Вид Документ Справка'. The editor area has a white background and displays a log file with four columns: '#data', 'counter', 'step', and 'time'. The data is organized into two groups, each starting with a header line. The first group starts with '#data counter step time' and contains 10 rows of data. The second group starts with '#data counter step time' and contains 17 rows of data. The data values are as follows:

#data	counter	step	time
0	1	3	10
1	2	7	503
0	3	10	521
0	4	13	643
1	5	15	726
0	6	18	1395
0	7	21	1505
0	8	24	1627
0	9	27	1826
0	10	30	2011
1	11	42	2322
1	12	45	2478
1	13	47	2570
1	14	50	2628
1	15	52	2646
1	16	56	2903
1	17	58	3009
1	18	60	3084
1	19	62	3096
1	20	64	3128
1	21	67	3274
1	22	69	3333
1	23	71	3356
1	24	75	3532
1	25	77	3535
0	26	79	3565

значения задержки в очереди превышали

График в какие периоды значения задержки в очереди превышали заданное

