Лабораторная работа 11

Модель системы массового обслуживания M |M |1

Акопян Сатеник

Содержание

1 Цель работы **5**

2 Задание **6**

3 Выполнение лабораторной работы **7**

4 Выводы **13**

Список литературы **14**

Список иллюстраций

3.1 Граф системы 7
3.2 Генератор заявок 8
3.3 Сервер обработки 8
3.4 Декларации системы 10
3.5 функция Predicate 11
3.6 функция Observer 12
3.7 Queue_Delay.log 12

Список таблиц

1 Цель работы

Построить модель системы массового обслуживания M |M |1

2 Задание

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

3 Выполнение лабораторной работы

1. Будем использовать три отдельных листа: на первом листе опишем граф системы (рис. 3.1), на втором — генератор заявок (рис. 3.2), на третьем — сервер обработки заявок (рис. 3.3), также зададим параметры модели на графах сети.

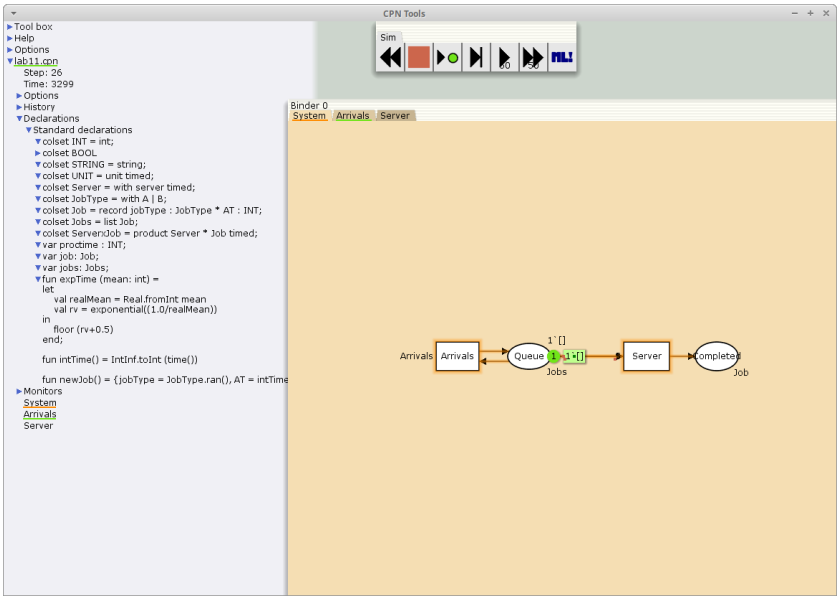

Рис. 3.1: Граф системы

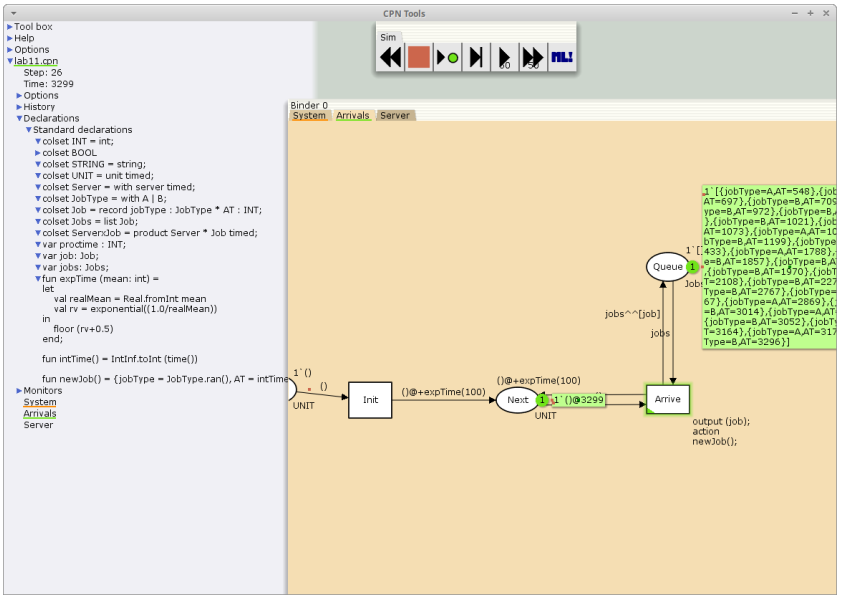

Рис. 3.2: Генератор заявок

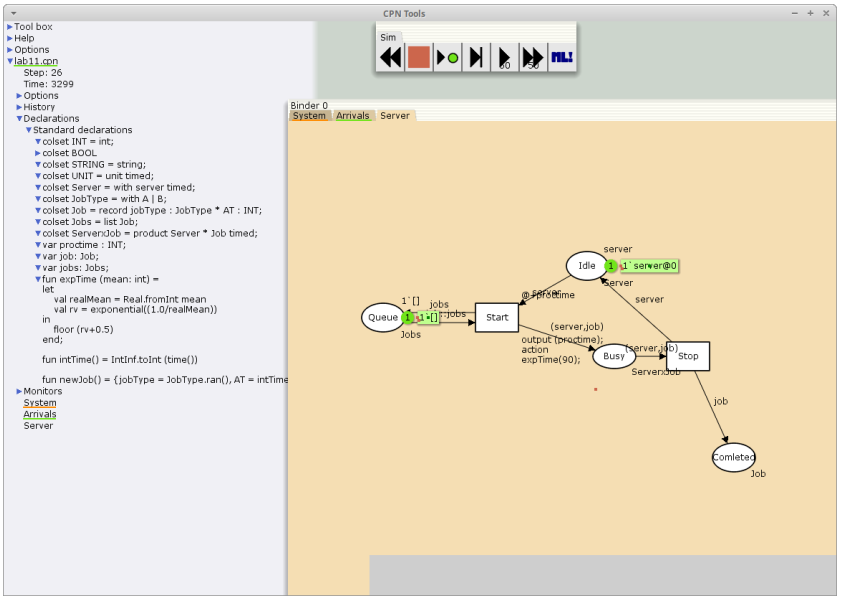

Рис. 3.3: Сервер обработки

2. Зададим декларации системы (рис. 3.4).

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.

Рис. 3.4: Декларации системы

3. Мониторинг параметров моделируемой системы

Необходимо внести изменения в функцию Predicate, которая будет выполнять-ся при запуске монитора

Изначально, когда функция начинает работать, она возвращает значение true,

в противном случае — false. В теле функции вызывается процедура predBindElem, которую определяем в предварительных декларациях. Зададим число шагов, через которое будем останавливать мониторинг. Для этого true заменим на Queue_Delay.count()=200 (рис. 3.5)

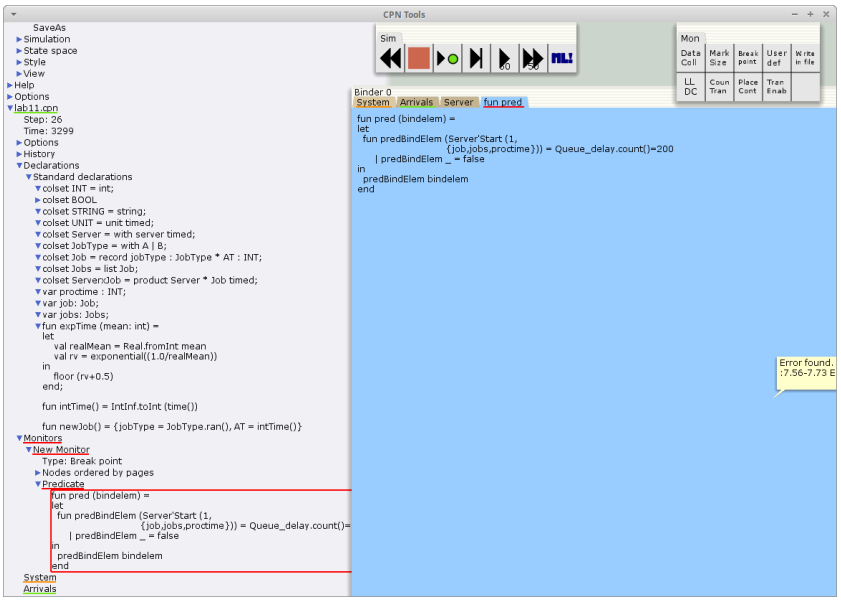

Рис. 3.5: функция Predicate

Изменим функцию Observer так, чтобы получить значение задержки в очереди. Для этого необходи- мо из текущего времени intTime() вычесть временную метку AT , означающую приход заявки в очередь (рис. 3.6)

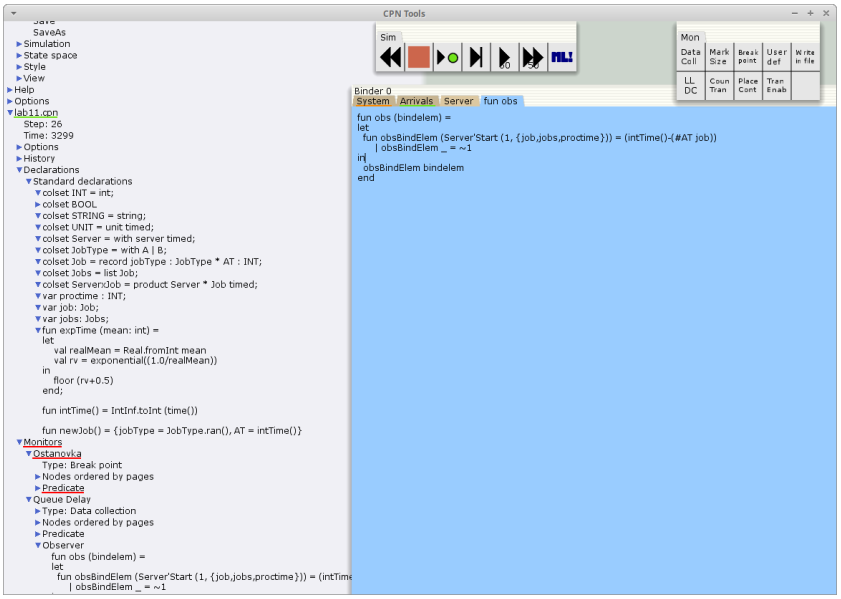

Рис. 3.6: функция Observer

После запуска программы на выполнение в каталоге с кодом программы появит- ся файл Queue_Delay.log, содержащий в первой колонке — значение задержки очереди, во второй — счётчик, в третьей — шаг, в четвёртой — время (рис. 3.7)

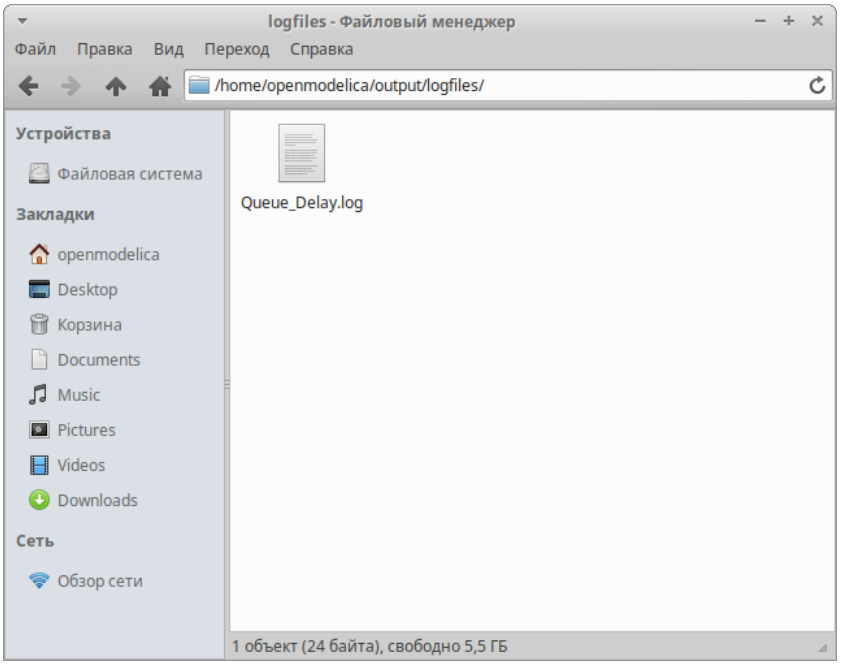

Рис. 3.7: Queue_Delay.log

4 Выводы

В результате была построена модель системы массового обслуживания M |M |1 с помощью cpntools

Список литературы