Лабораторная работа 11

Модель системы массового обслуживания M |M |1

Акопян Сатеник

Содержание

Список иллюстраций

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. 1), на втором — генератор заявок (рис. 2), на третьем — сервер обработки заявок (рис. 3), также зададим параметры модели на графах сети.

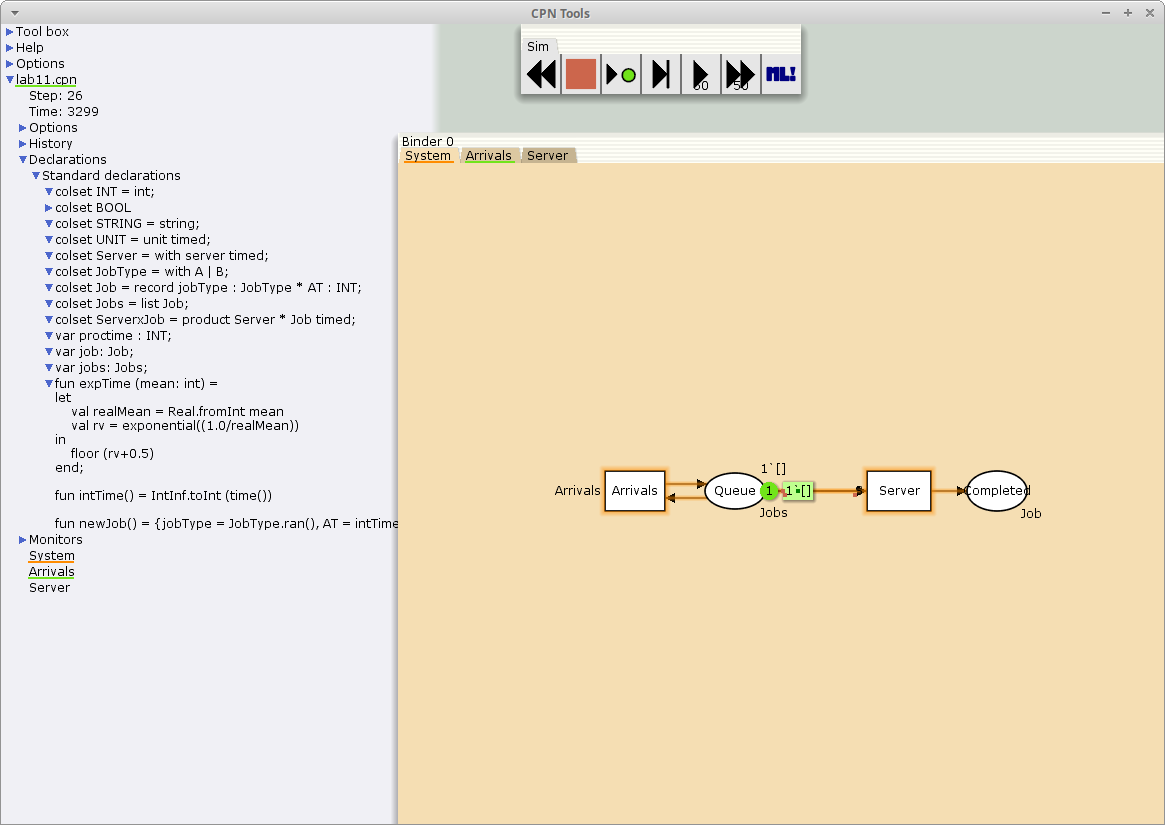


Рис. 1: Граф системы

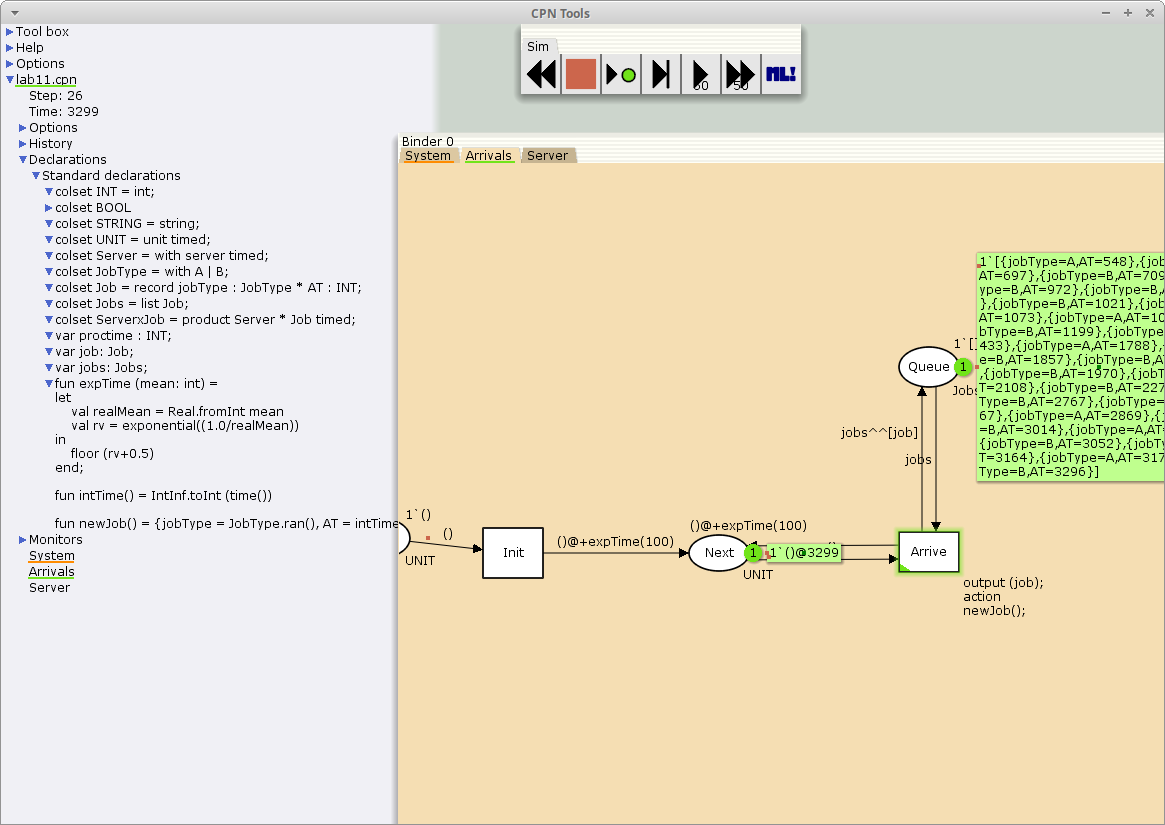


Рис. 2: Генератор заявок

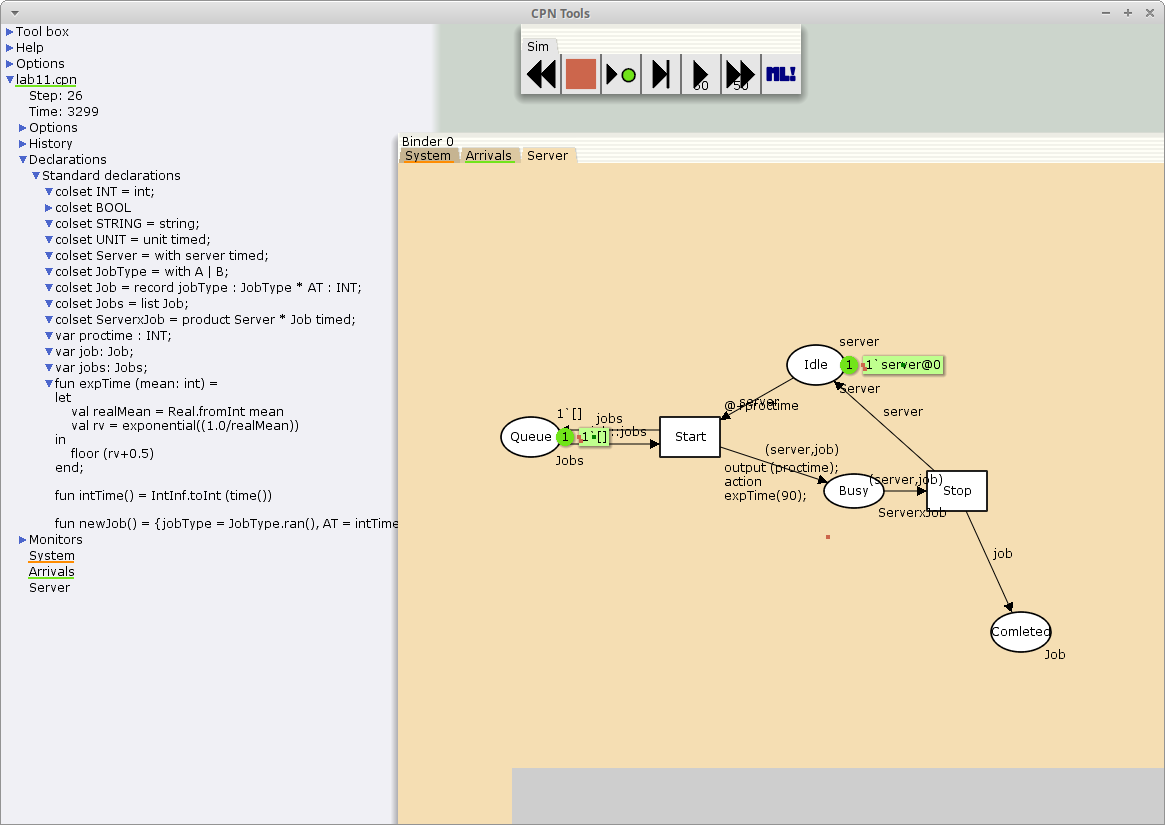


Рис. 3: Сервер обработки

1. Зададим декларации системы (рис. 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 име- ет 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.

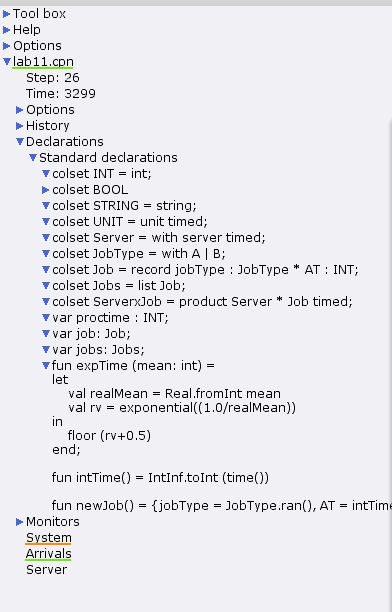


Рис. 4: Декларации системы

1. Мониторинг параметров моделируемой системы

Необходимо внести изменения в функцию Predicate, которая будет выполняться 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. 5)

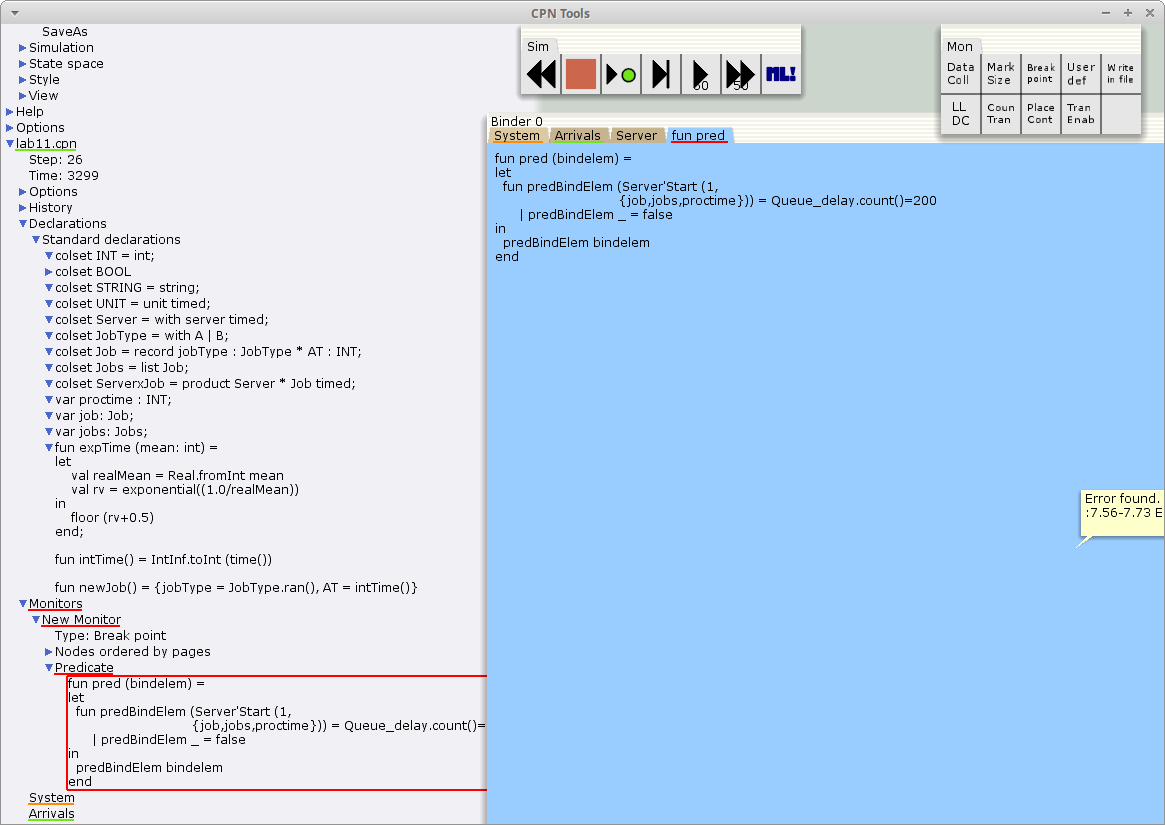


Рис. 5: функция Predicate

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. 6)

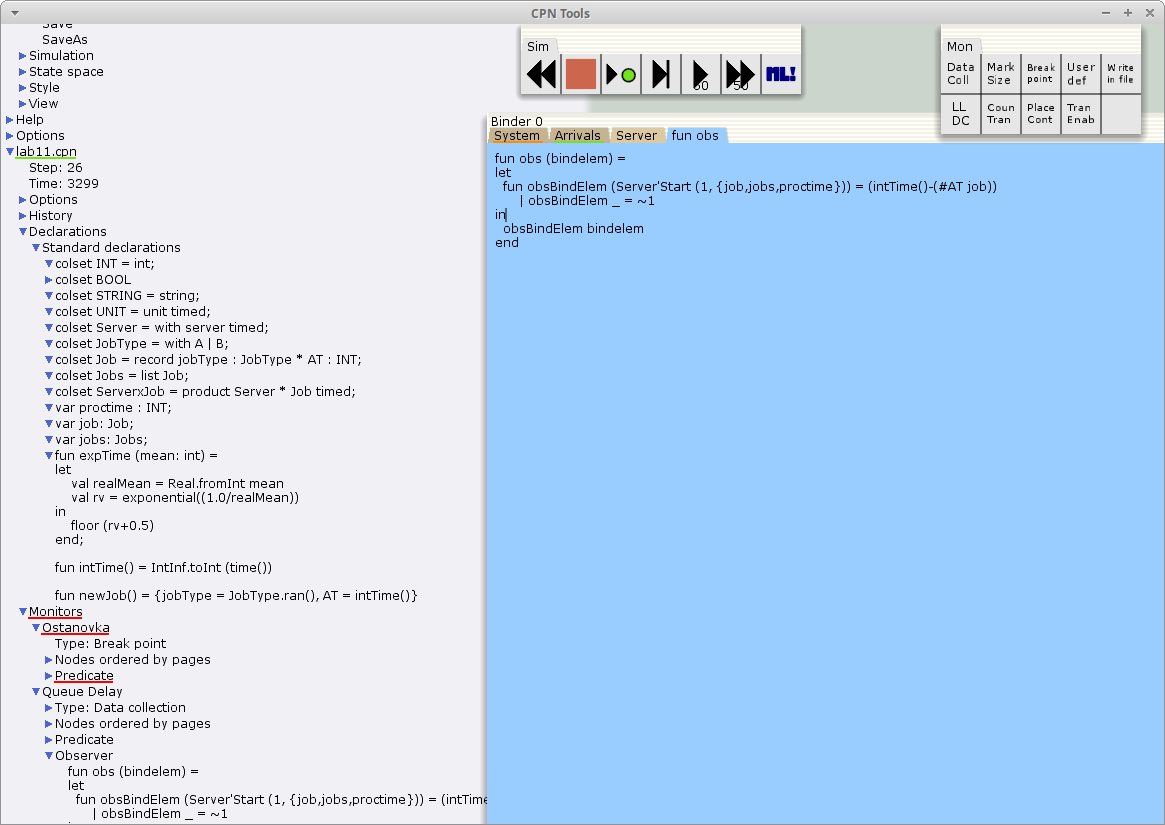


Рис. 6: функция Observer

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. 7)

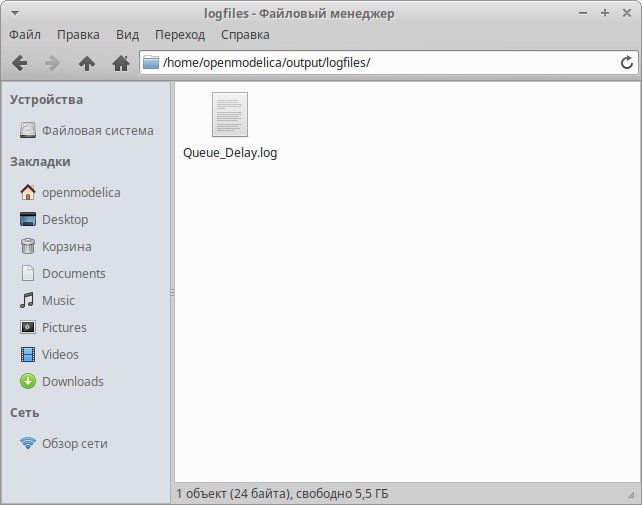


Рис. 7: Queue\_Delay.log

# 4 Выводы

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

# Список литературы