

Лабораторная работа № 7

Модель СМО

Демидова Екатерина Алексеевна

Содержание

1	Введение	4
1.1	Цели и задачи	4
2	Выполнение лабораторной работы	5
2.1	Реализация модели в xcoss	5
3	Выводы	10

Список иллюстраций

2.1	Переменное окружение	5
2.2	Суперблок, моделирующий поступление заявок	6
2.3	Суперблок, моделирующий обработку заявок	7
2.4	Модель $M M 1 \infty$ в xcos	8
2.5	Поступление(зеленым) и обработка(черным) заявок	9
2.6	Динамика размера очереди	9

1 Введение

1.1 Цели и задачи

Цель работы

Реализовать модель $M|M|1|\infty$ с помощью xcos.

Задание

- Реализовать в xcos модель системы массового обслуживания типа $M|M|1|\infty$.
- Построить график, описывающий динамику размера очереди
- Построить график, описывающий поступление и обработку заявок.

2 Выполнение лабораторной работы

2.1 Реализация модели в хcos

В нашей модели одна очереди, поступление заявок описывается пуассоновским процессом.

Зададим переменное окружение(рис. [2.1]).

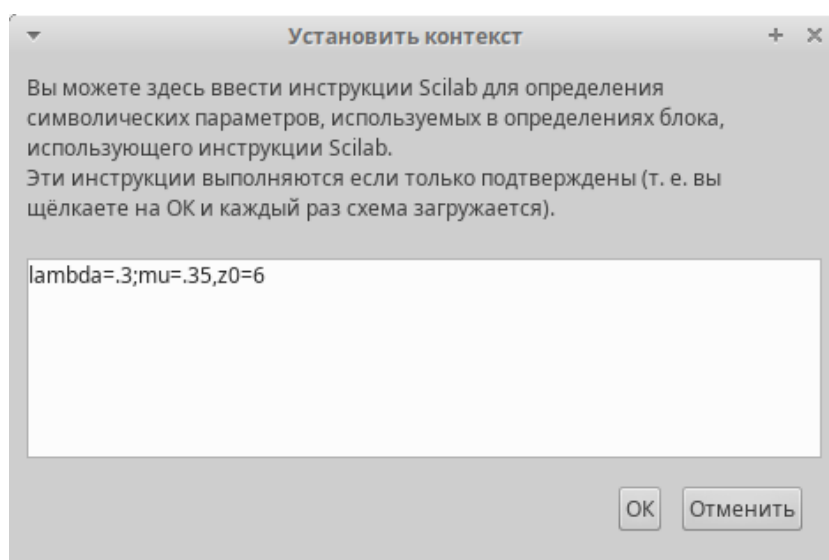


Рис. 2.1: Переменное окружение

Далее приведены блоки и их связь для моделирования рассматриваемой системы.

В нашей модели есть суперблок для описания поступления заявок(рис. [2.2]):

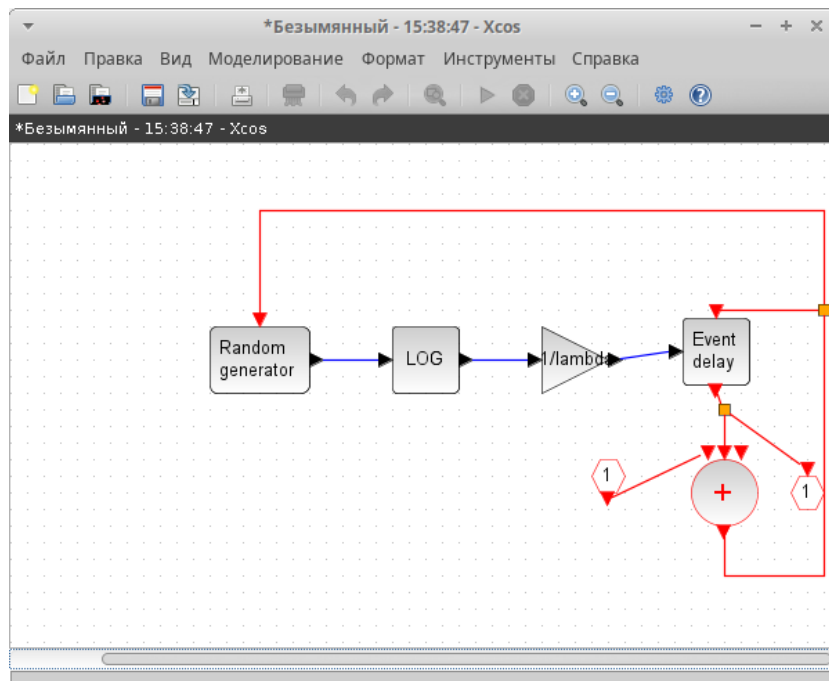


Рис. 2.2: Суперблок, моделирующий поступление заявок

- RAND_M – генератор случайных чисел по равномерному распределению.
- LOGBLCK_f – взятие логарифма от потока выхода случайных чисел, чтобы получить Пуассоновское распределение.
- GAINBLCK_f – умножает сгенерированный поток по Пуассоновскому распределению на $-\frac{1}{\lambda}$
- EVTGEN_f – обработчик событий, так как для моделирования заявок будут использованы события.
- CLKSOMV_f – синхронизация выходных и входных сигналов.
- CLKINV_f – порт входа в суперблок.
- CLKOUTV_f – порт выхода из суперблок.

Также есть суперблок, описывающий обработку заявок(рис. [2.3]):

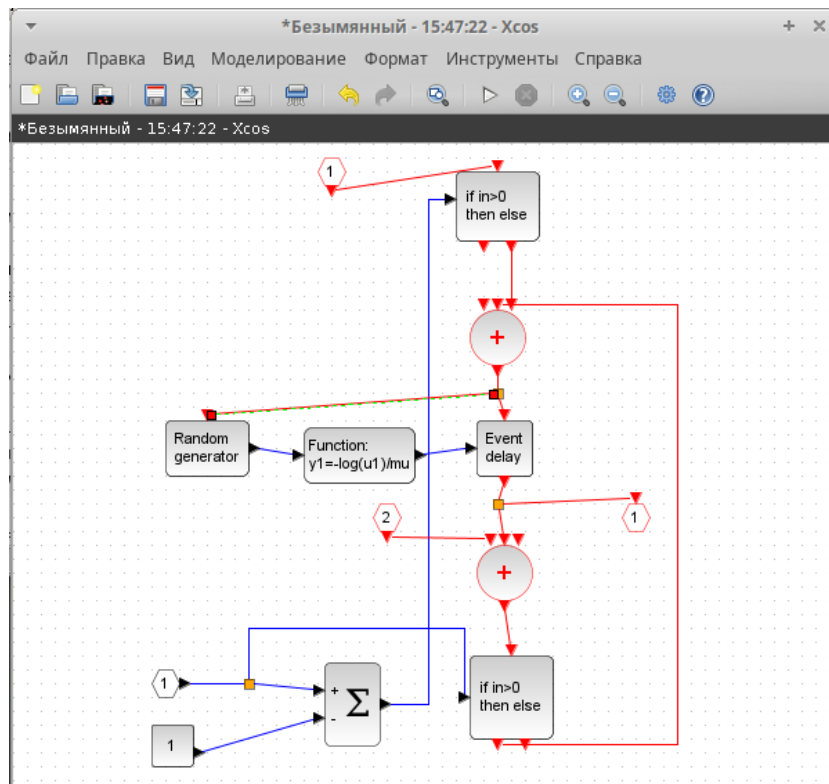


Рис. 2.3: Суперблок, моделирующий обработку заявок

- RAND_M – генератор случайных чисел по равномерному распределению.
- sci_funk_m_block – задает математическое выражение $y1 = -\log(u1)/\mu$, которое ранее мы задавали блоками.
- EVTGEN_f – обработчик событий, так как для моделирования заявок будут использованы события.
- CLKSOMV_f – синхронизация выходных и входных сигналов. В этом супер-блоке их два.
- IFHEL_f – два блока для определения длины очереди, если значение больше нуля, то сигнал подается.
- CLKINV_f – входы для запуска и для сообщения о том, что сообщение пришло в очередь, чтобы по разному обрабатывать пустую и не пустую очередь.
- IN_f, CONST_M – проверка на длину очереди

Вся модель выглядит следующим образом(рис. [2.4]):

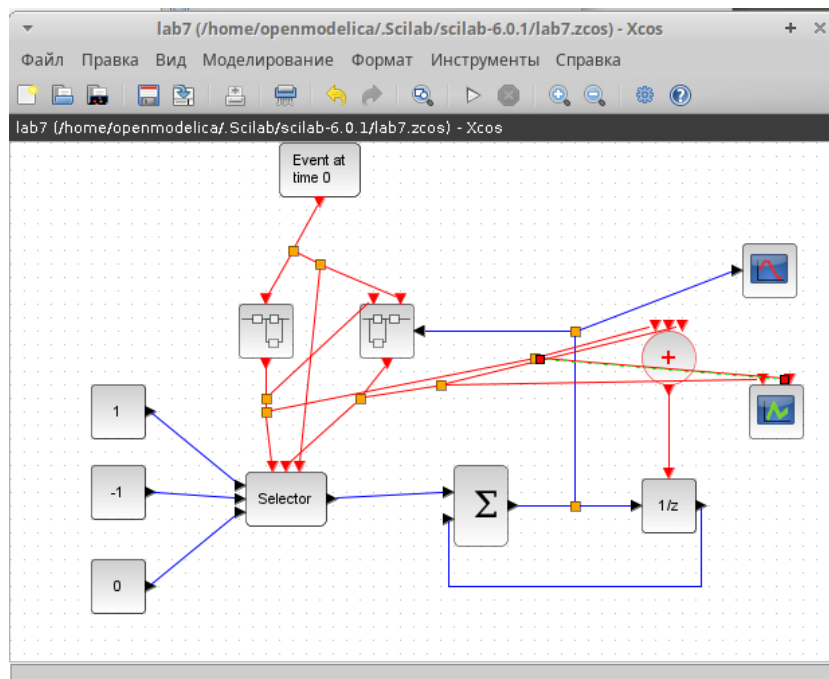


Рис. 2.4: Модель $M|M|1|\infty$ в xcos

- SELECTOR_M – берёт входные сигналы и с помощью управляющих сигналов будет добавлять вход к очереди, либо считывать. У него три входа – для поступления заявок, обработки заявок и начальной синхронизации.
- CONST_M – поступление заявки выражается 1, обслуживание заявки – -1, первоначальная синхронизация – 0.
- EVTGEN_f – запуск первоначального события в нулевой момент времени.
- DOLLAR_f – блок для имитации очереди, на него приходит управление, которое синхронизируется с источника и с обработчика.
- CSCOPE – для отрисовки длины очереди.
- CEVEBTSCOPE – обработка событий.

В результате получим два графика: один показывает поступление и обработку заявок, а второй изменение длины очереди(рис. [2.5], [2.6]).

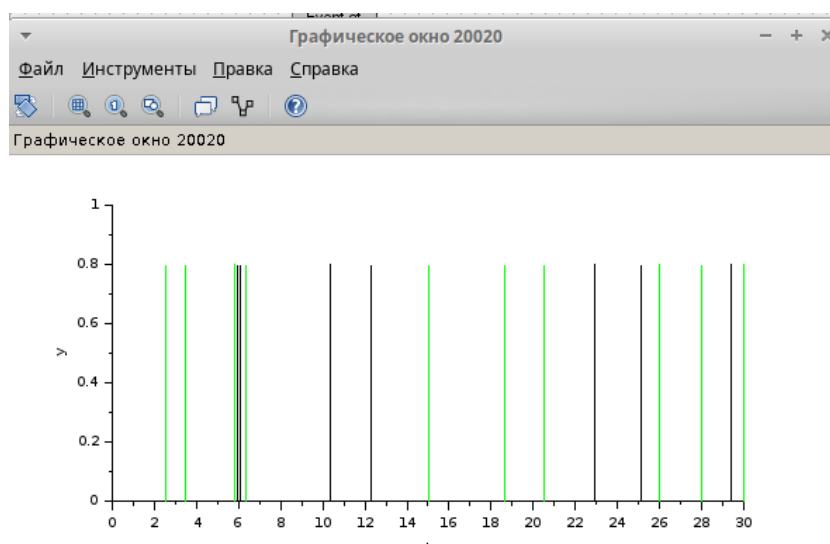


Рис. 2.5: Поступление(зеленым) и обработка(черным) заявок

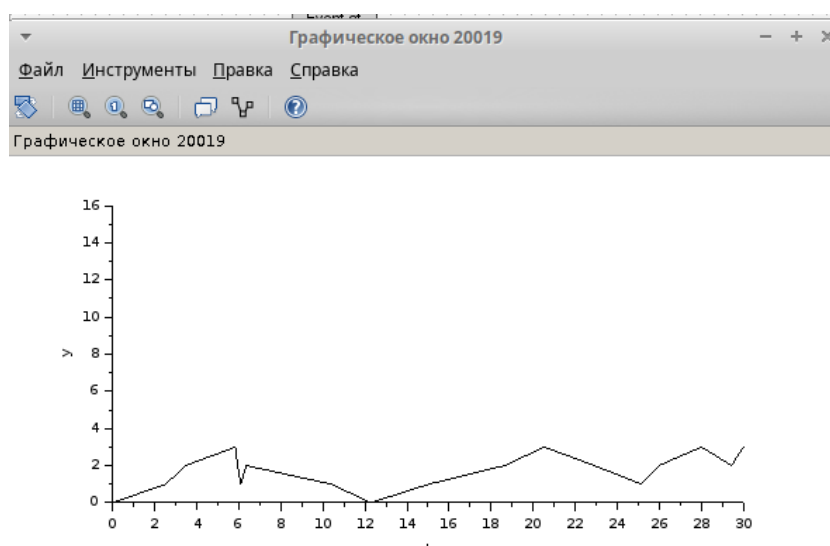


Рис. 2.6: Динамика размера очереди

3 Выводы

В результате выполнения работы была реализована модель $M|M|1|\infty$ с помощью xcos.