Сапегин Павел Александрович, группа 7-2

Лабораторная работа № 3

**Вариант № 3**

Моделирование систем массового обслуживания (SimEvents)

**Цель работы**

Практическое изучение технологий визуального программирования имитационных моделей систем с использованием подсистемы Simulink, построение систем массового обслуживания, а также оценка различных показателей эффективности с помощью библиотеки SimEvents.

**Задание**

3. Междугородный переговорный пункт имеет четыре телефонных аппарата. В среднем за сутки поступает 320 заявок на переговоры. Средняя длительность переговоров составляет 5 мин. Длина очереди не должна превышать 6 абонентов. Потоки заявок и обслуживаний простейшие. Определить характеристики обслуживания переговорного пункта в стационарном режиме: вероятность простоя каналов, вероятность отказа клиенту в обслуживании, среднее число занятых каналов, среднее число заявок в очереди

**Код файла .m**

Код приложен целиком, так как был написан с нуля.

% 3. Междугородный переговорный пункт имеет четыре телефонных

% аппарата. В среднем за сутки поступает 320 заявок на переговоры. Средняя

% длительность переговоров составляет 5 мин. Длина очереди не должна

% превышать 6 абонентов. Потоки заявок и обслуживаний простейшие.

% Определить характеристики обслуживания переговорного пункта в

% стационарном режиме:

% вероятность простоя каналов,

% вероятность отказа клиенту в обслуживании,

% среднее число занятых каналов,

% среднее число заявок в очереди.

clear all; % очищаем рабочую область

Ts = 0.25; % снимаем показания каждые 15 секунд (период дискретизации)

Ns = 60 \* 24; % отчет на протяжении 24 часов

sim('model', Ts \* Ns); % запуск моделирования

% 1 ПАРАМЕТР: Считаем время занятости каналов

channel1\_sum = sum(channel1);

channel2\_sum = sum(channel2);

channel3\_sum = sum(channel3);

channel4\_sum = sum(channel4);

% Получаем количество отсчетов для каждого канала

channel1\_len = length(channel1);

channel2\_len = length(channel2);

channel3\_len = length(channel3);

channel4\_len = length(channel4);

% Рассчитываем для каждого канала время простоя

p\_downtime1 = 1 - channel1\_sum / channel1\_len;

p\_downtime2 = 1 - channel2\_sum / channel2\_len;

p\_downtime3 = 1 - channel3\_sum / channel3\_len;

p\_downtime4 = 1 - channel4\_sum / channel4\_len;

fprintf('Время простоя каналов: %f, %f, %f, %f\n', ...

p\_downtime1, p\_downtime2, p\_downtime3, p\_downtime4); % выводим в консоль

% 2 ПАРАМЕТР: Считаем вероятность отказа клиенту

probability\_of\_failure = 1 - (sum(entitiesNumFromQueue) ...

/ sum(enitiesNumFromGenerator));

fprintf('Вероятность отказа клиенту в обслуживании: %f\n', ...

probability\_of\_failure); % Выводим в консоль вероятность отказа

% 3 ПАРАМЕТР: Рассчитываем среднее число занятых каналов

n = length(channel1); % длина массивов

active\_counts = zeros(1, n); % сюда будем сохранять количество ненулевых значений

% Запускаем цикл по длине массивов

for i = 1:n

% Проверяем, какие из 4 каналов ненулевые в текущем индексе

count = (channel1(i) ~= 0) + ...

(channel2(i) ~= 0) + ...

(channel3(i) ~= 0) + ...

(channel4(i) ~= 0);

active\_counts(i) = count; % сохраняем, сколько каналов активно

end

% Вычисляем среднее количество одновременно активных каналов

average\_active\_channels = mean(active\_counts);

fprintf('Среднее число активных каналов: %f\n', ...

average\_active\_channels); % Выводим в консоль

% Считаем среднее кол-во заявок в очереди

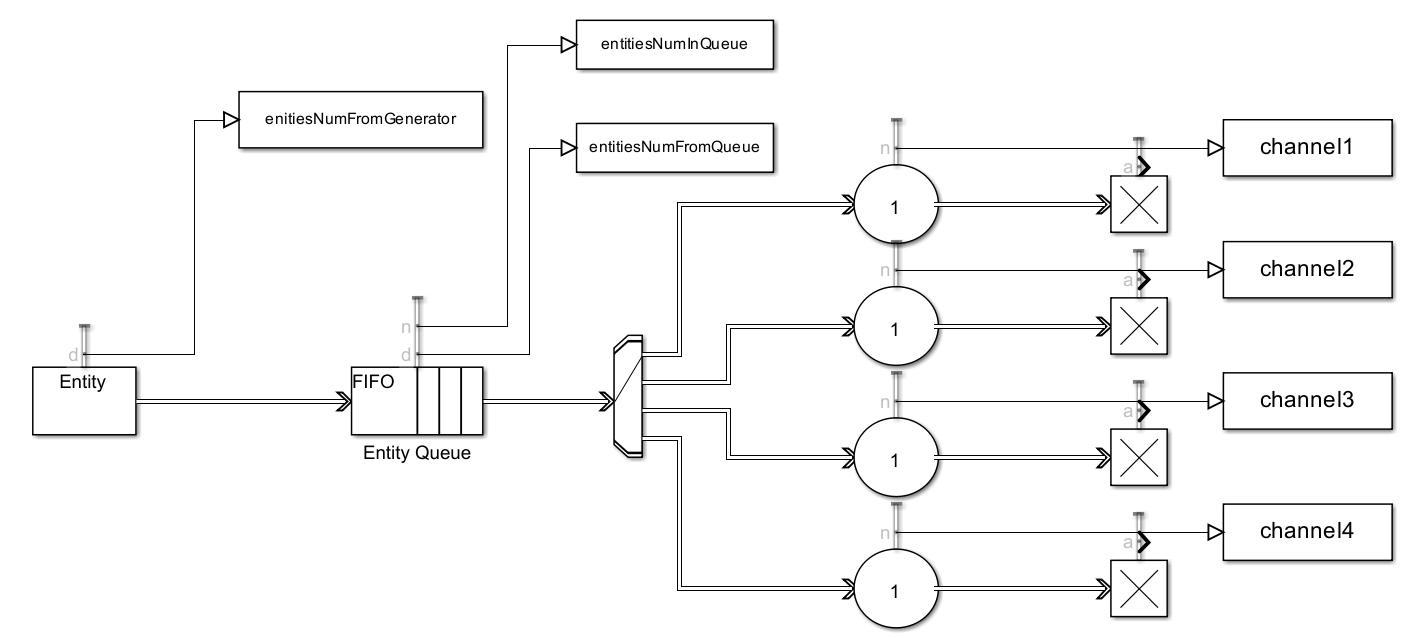
average\_number\_of\_requests\_in\_queue = sum(entitiesNumInQueue) / length(entitiesNumInQueue);

fprintf('Среднее число заявок в очереди: %f\n', ...

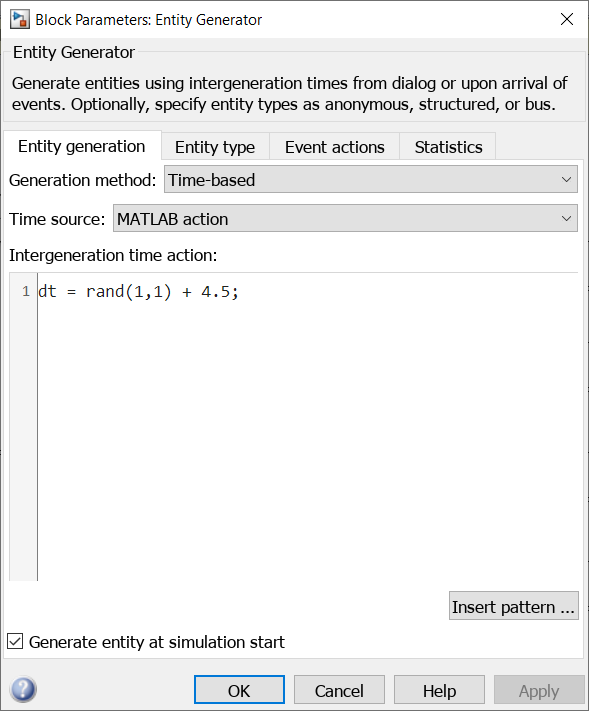
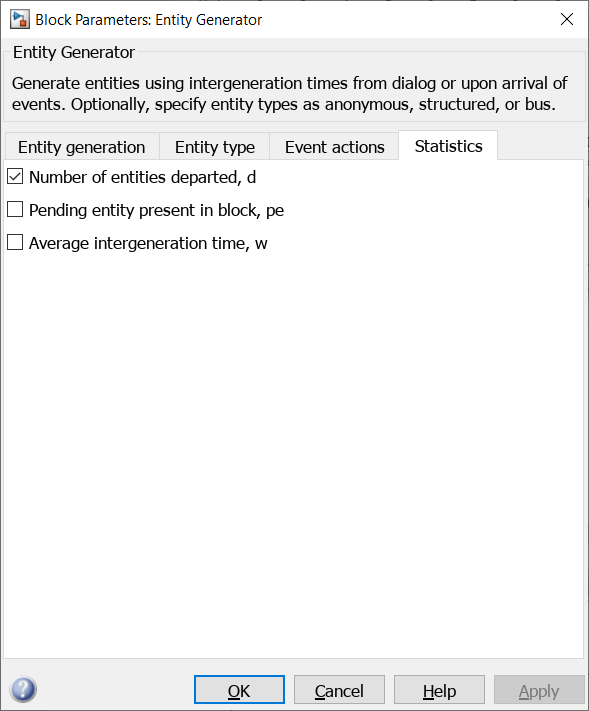
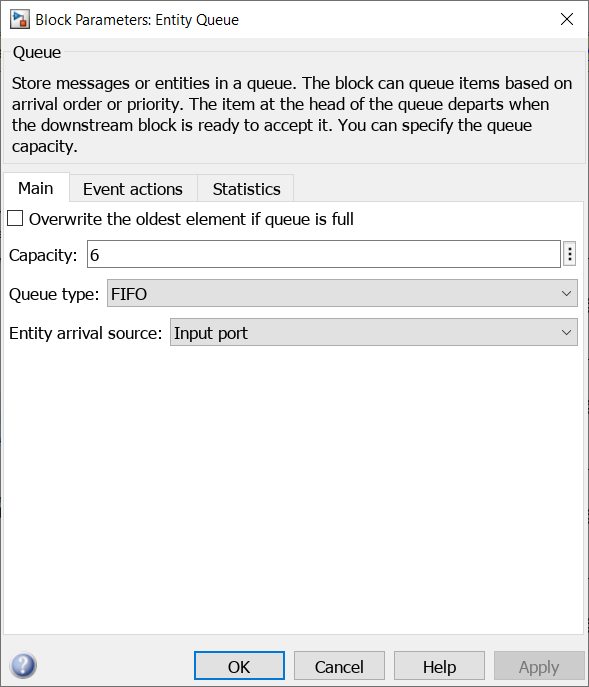
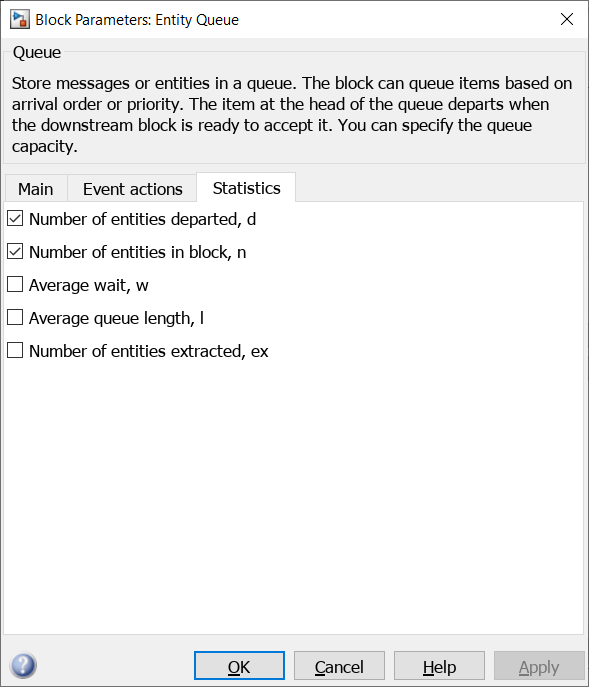
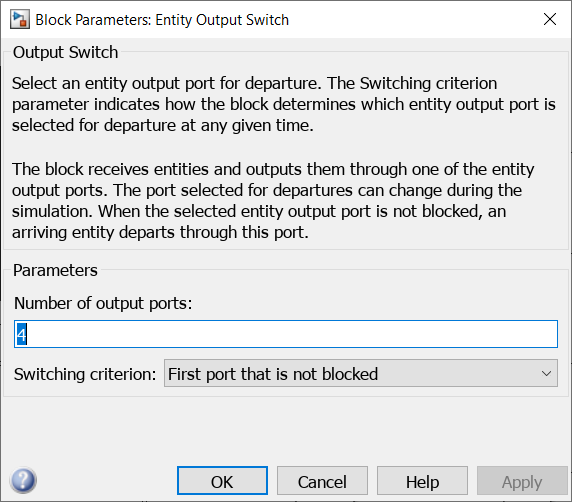
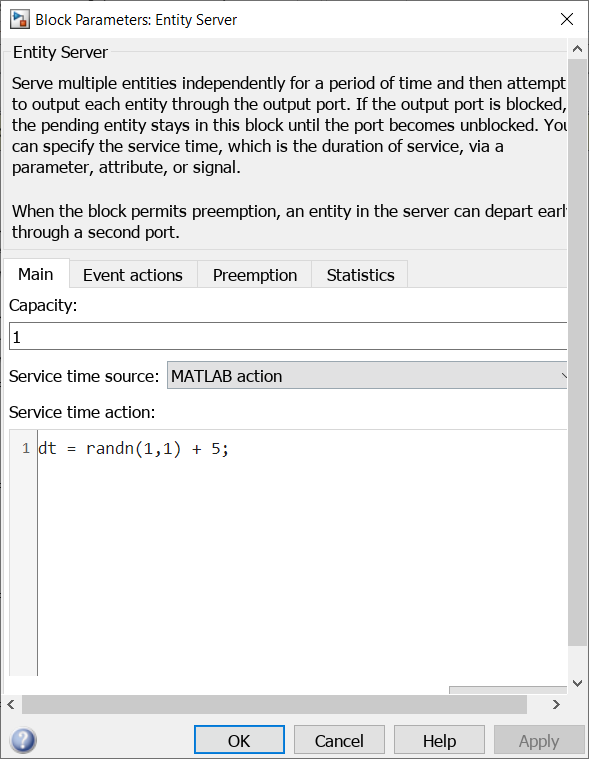
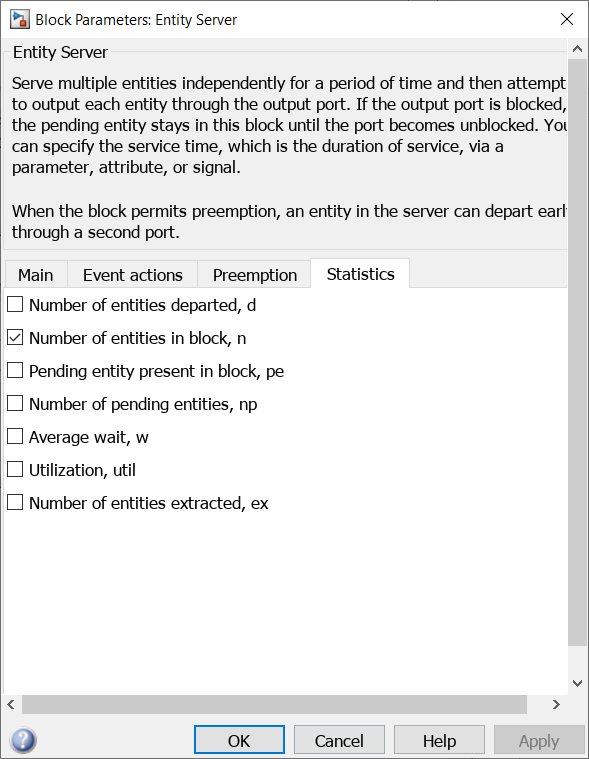
average\_number\_of\_requests\_in\_queue); % Выводим в консоль

**Настройки блоков и графики**

Скриншот реализованной модели:



Параметры блоков:

1. Параметры блока «Генератор сущностей»:  
     
   
2. Параметры блока «Очередь сущностей»  
     
   
3. Параметры блока «Output Switch»  
   
4. Параметры блока «Сервер» (в модели 4 сервера; они все одинаковые)  
     
   
5. Также в модели используются блоки «To Workspace». Для всех таких блоков параметр «Sample Time» записана переменная Ts.

**Результаты выполнения задания**

В результате выполнения задания были оценены параметры эффективности, величины которых видно на следующем скриншоте:

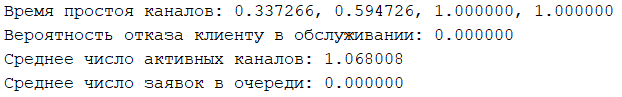


Рисунок 1 – Результат выполнения скрипта

По полученным результатам видно, что в данной симуляции каналы загружены не целиком, следовательно, при тех же параметрах система может обслуживать большее количество клиентов.

**Выводы**

Была реализована модель в Симулинке, которая имитирует обслуживание клиентов переговорного пункта.

В управляющей программе (.m файл) реализован подсчет заданных показателей эффективности:

* вероятность простоя каналов,
* вероятность отказа клиенту в обслуживании,
* среднее число занятых каналов,
* среднее число заявок в очереди.

Посчитанные показатели эффективности печатаются в консоль как результат выполнения программы.

В процессе выполнения задания возникла проблема: блоки «To Workspace» не передавали данные в рабочую область. Суть проблемы заключалась в том, что эти данные были упакованы в переменную out. Для решения в настройках переменная out была отключена:

