Лабораторная работа №5

«СМО с ограниченной очередью»

На сервере есть n каналов передачи сообщений. Среднее время обработки сообщений τ. На сервер поступают сообщения в среднем количестве λ сообщений в минуту. Для хранения сообщений в очереди на сервере выделено место для (n+1) сообщения. Определить основные характеристики сервера - вероятность очереди, среднее число занятых каналов, средняя длина очереди, среднее число сообщений на сервере; предполагается, что сообщение не получает отказ при занятости всех каналов и очереди длины не больше (n+1). Параметры варианта определяются по формулам: n=3+[(i+j)/8], λ=1+i/4, τ=5/(5+j). Здесь квадратные скобки означают взятие целой части, а i,j - последние цифры зачетки.

Исходные данные варианта

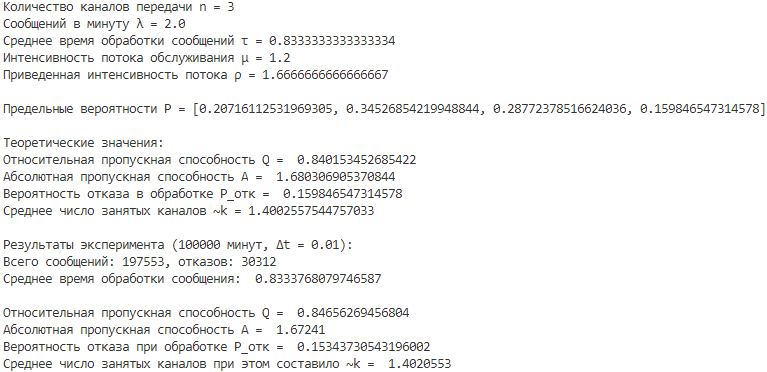
Ход работы

1. Вычислим интенсивность потока обслуживания и приведённую интенсивность потока:
2. Вычислим предельные вероятности:
3. Вычислим относительную и абсолютную пропускную способность, вероятность отказа и среднее число занятых каналов:
4. Реализуем программу для практической проверки полученных данных.

Программа имитирует работу СМО заданное время с заданными интервалами времени, на основе которых определяются дискретные состояния системы.

Вывод: в ходе лабораторной работы были вычисленные такие характеристики СМО, как относительная и абсолютная пропускная способность, вероятность отказа и среднее число занятых каналов. Также была разработана программа для проведения практического эксперимента. В результате работы программы были получены схожие, с точностью до тысячных, данные.

Пример работы программы



Код программы

import random

import math

import numpy

from scipy.special import factorial as fact

i = 4

j = 1

n = 3 + ((i + j) // 8)          # Количество каналов передачи сообщений

lambd = 1 + i / 4               # Интенсивность потока заявок

tau = 5 / (5 + j)               # Среднее время обработки сообщений

print("Количество каналов передачи n = {}\nСообщений в минуту λ = {}\nСреднее время обработки сообщений τ = {}".format(n, lambd, tau))

mu = 1 / tau                    # Интенсивность потока обслуживания

ro = lambd / mu                 # Приведенная интенсивность потока заявок (интенсивность нагрузки)

print("Интенсивность потока обслуживания μ = {}\nПриведенная интенсивность потока ρ = {}\n".format(mu, ro))

P = [0]                         # Предельные вероятности (среднее относительное время, которое канал занят (p0 - все свободны))

for i in range (0, n + 1):

     P[0] += (ro \*\* i) / fact(i)

P[0] = P[0] \*\* -1

for i in range (1, n + 1):

     P.append((ro \*\* i) / fact(i) \* P[0])

print("Предельные вероятности P = {}\n".format(P))

P\_o = (ro \*\* n) / fact(n) \* P[0]                # Вероятность отказа (все каналы заняты)

Q = 1 - P\_o                                     # Относительная пропускная способность

A = lambd \* Q                                   # Абсолютная пропускная способность

k = 0                                           # Среднее число занятых каналов

for i in range(0, n + 1):

     k += i \* P[i]

print("Теоретические значения:")

print("Относительная пропускная способность Q = ", Q)

print("Абсолютная пропускная способность A = ", A)

print("Вероятность отказа в обработке P\_отк = ", P\_o)

print("Среднее число занятых каналов ~k = {}\n".format(k))

# Возвращает номер свободного канала, иначе - -1

def freeChannel(channels):

     for i in range(0, n):

          if channels[i] == 0:

               return i

     return -1

# Обработка уже имеющихся сообщений

def messagesProcessing(channels, dt):

     for i in range (0, len(channels)):

          if (channels[i] > dt):

               channels[i] -= dt

          else:

               channels[i] = 0

# Получение нового сообщения

def newMessage(message, channels):

     pos = freeChannel(channels)

     if (pos != -1):

          channels[pos] = message

          return True

     return False

channels = [0 for i in range (0, n)] # Каналы связи (0, если не обрабатывается, иначе - оставшееся время)

maxTime = 100000         # Время работы

busyChannels = 0         # Занятые каналы

unProcessedMessages = 0  # Необработанные сообщения

totalMessages = 0        # Всего сообщений

dt = 0.01                # Δt

averageTime = 0          # Среднее время обработки сообщения

for currentTime in range (0, int(maxTime / dt)):

     if (random.random() < 1 - math.exp(-1 \* lambd \* dt)):       # Если сообщение пришло

          message = tau - 0.05 + random.random() / 10            # Назначаем ему время обработки

          averageTime += message

          totalMessages += 1

          if(newMessage(message, channels) != True):             # Отправляем сообщение в свободный канал

               unProcessedMessages += 1                          # Если все каналы заняты, сообщение не обработано

     busyChannels += n - channels.count(0)

     messagesProcessing(channels, dt)                            # Обрабатываем сообщение

busyChannels = busyChannels / (maxTime / dt)

averageTime = averageTime / totalMessages

print("Результаты эксперимента ({} минут, Δt = {}):".format(maxTime, dt))

print("Всего сообщений: {}, отказов: {}".format(totalMessages, unProcessedMessages))

print("Среднее время обработки сообщения: ", averageTime)

print("\nОтносительная пропускная способность Q = ", (totalMessages - unProcessedMessages) / totalMessages)

print("Абсолютная пропускная способность А = ", (totalMessages - unProcessedMessages) / maxTime)

print("Вероятность отказа при обработке P\_отк = ", unProcessedMessages / totalMessages)

print("Среднее число занятых каналов при этом составило ~k = ", busyChannels)