**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ “ЛЭТИ”**

**ИМЕНИ В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ИС**

**Отчет**

**по Курсовой Работе**

**по дисциплине “Моделирование Систем Массового Обслуживания”**

**Вариант №29**

Студент гр.1374 Дюков Н.В.

Преподаватель Цехановский В.В.

Санкт-Петербург

2024

**Задание на курсовую работу**

Студент: Дюков Николай

Группа: 1374

Тема работы: Моделирование системы массового обслуживания

Исходные данные:

В специализированной вычислительной системе периодически выполняются три вида заданий, которые характеризуются уровнями приоритета: нулевым, первым и вторым. Каждый новый запуск задания оператор производит с помощью дисплея, работая на нем 50 ± 30 с. После запуска задания оно требует для своего выполнения 100 ± 50 с времени работы процессора, причем задания более высокого приоритета прерывают выполнение задач более низкого приоритета. Результаты обработки задания выводятся на печать без прерываний в течение 30 ± 10 с, после чего производится их анализ в течение 60 ± 20 с, и задание запускается снова. Можно считать, что при работе дисплея и при выводе результатов на печать процессор не используется. Смоделировать процесс работы системы при условии, что задание второго уровня приоритета выполняется 100 раз. Подсчитать число циклов выполнения остальных задний и определить коэффициенты загрузки технических средств системы.

Содержание пояснительной записки:

«Введение», «Построение концептуальной модели», «Разработка имитационной модели», «Проведение экспериментов», «Анализ результатов».

Дата выдачи задания: 01.09.2024

Дата сдачи задания: 25.12.2024

|  |  |  |
| --- | --- | --- |
| Студент гр. 1374 |  | Дюков Н.В. |
| Преподаватель |  | Цехановский В.В. |

# **Содержание**

Оглавление

[**Содержание** 3](#_Toc185100803)

[**Введение** 4](#_Toc185100804)

**1.ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ**………………………………………………………………………5

[**1.1. Анализ задачи.** 5](#_Toc185100805)

[**1.2. Формализация: обоснование применения Q-схем.** 6](#_Toc185100807)

[**2. РАЗРАБОТКА МОДЕЛИРУЮЩЕГО АЛГОРИТМА: БЛОК-СХЕМА ОБОБЩЕННОГО АЛГОРИТМА ИМИТАЦИИ, БЛОК-СХЕМЫ ЧАСТНЫХ АЛГОРИТМОВ ИМИТАЦИИ.** 7](#_Toc185100808)

[**3. РАЗРАБОТКА МОДЕЛИРУЮЩЕЙ ПРОГРАММЫ** 8](#_Toc185100809)

[**4. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА** 14](#_Toc185100810)

[**5. АНАЛИЗ РЕЗУЛЬТАТОВ** 17](#_Toc185100811)

[**5.1. Моделирование улучшенного варианта системы.** 17](#_Toc185100812)

[**5.2. Интерпретация результатов и рекомендации по реализации моделируемой системы.** 18](#_Toc185100813)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 21](#_Toc185100814)

[**ПРИЛОЖЕНИЕ А** 22](#_Toc185100815)

# **Введение**

В данной работе будет выполнено имитационное моделирование системы массового обслуживания. Мы подробно рассмотрим структурную и Q-схемы системы, а также разработаем программу, которая будет осуществлять имитацию функционирования сети. Программа предоставит результаты, которые помогут понять текущее состояние системы, а также выявить возможные недостатки, что позволит рассмотреть пути её оптимизации, включая корректировку параметров, таких как время обслуживания.

Имитационное моделирование будет реализовано с использованием языка программирования GPSS в среде GPSS-World. Полный код программы можно найти в приложении A.**1. ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ**

## **1.1. Анализ задачи.**

Вариант 29. В специализированной вычислительной системе периодически выполняются три вида заданий, которые характеризуются уровнями приоритета: нулевым, первым и вторым. Каждый новый запуск задания оператор производит с помощью дисплея, работая на нем 50 ± 30 с. После запуска задания оно требует для своего выполнения 100 ± 50 с времени работы процессора, причем задания более высокого приоритета прерывают выполнение задач более низкого приоритета. Результаты обработки задания выводятся на печать без прерываний в течение 30 ± 10 с, после чего производится их анализ в течение 60 ± 20 с, и задание запускается снова. Можно считать, что при работе дисплея и при выводе результатов на печать процессор не используется. Смоделировать процесс работы системы при условии, что задание второго уровня приоритета выполняется 100 раз. Подсчитать число циклов выполнения остальных задний и определить коэффициенты загрузки технических средств системы.

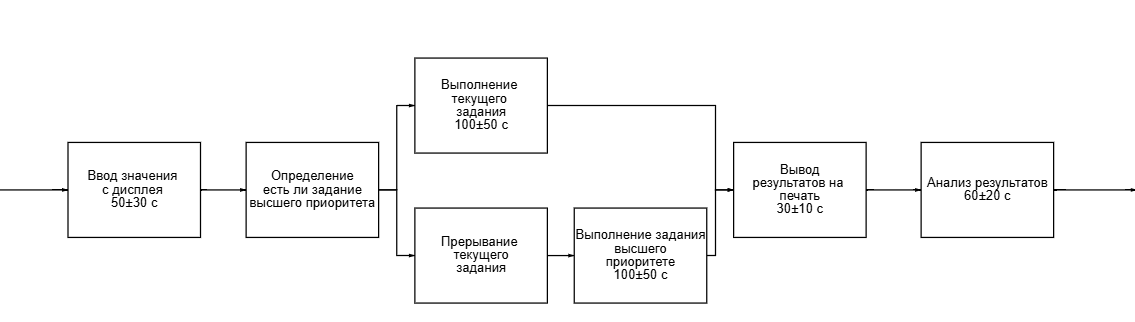
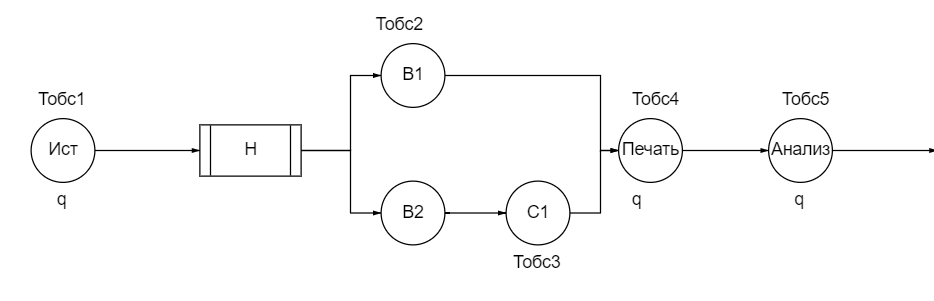
  
рис.1 Структурная схема модели

Схема моделирует процесс обработки заданий с тремя уровнями приоритета. Блок "Определение есть ли задание высшего приоритета" явно указывает на наличие механизма, который проверяет, существует ли задание с более высоким приоритетом, чем текущее. Если такое задание есть, текущее задание прерывается, и процессор начинает обрабатывать задание с более высоким приоритетом. Это ключевое свойство приоритетных очередей. Задания с более низким приоритетом ожидают своей обработки в неявной очереди. Порядок обработки определяется приоритетом, а не временем поступления (как в FIFO). Поэтому используем приоритетную очередь.

## **1.2. Формализация: обоснование применения Q-схем.**

Так как рассматриваемый процесс обслуживания абонентов по своей сути является процессом, то для его формализации воспользуемся аппаратом Q-схем. Исследуемая система является сетью массового обслуживания (здесь и далее – СеМО. Построенную Q-схему изобразим на рис. 2:

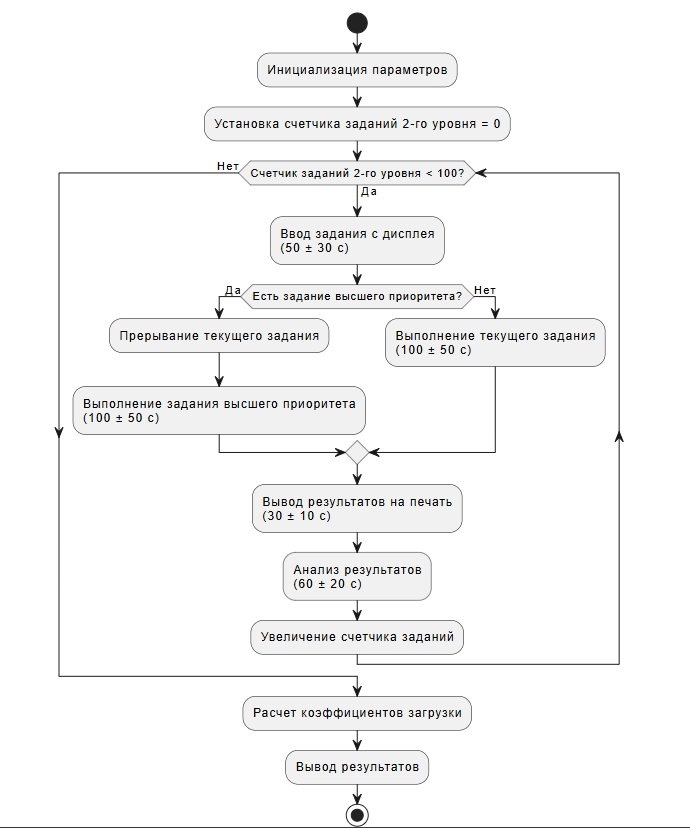
  
рис.2 Q-схема модели

Источник – дисплей, работая на нем 50 ± 30 с.

Н - накопитель  
В1, С1 – выполнение текущего задания/ с высшим приоритетом, 100 ± 50 с.  
В2 – прерывание текущего задания  
Печать - результаты обработки задания выводятся на печать без прерываний в течение 30 ± 10 с.  
Анализ - производится анализ в течение 60 ± 20 с.  
Тобс1, Тобс2, Тобс3, Тобс4, Тобс5 – время обслуживания заявки в данном состоянии.

Мы будем исследовать в данной работе коэффициент загрузки q.

Поскольку использование аналитических расчётов не имеет смысла для данной системы (так как это не экспоненциальная СеМО и имеется дополнительное условие на то, каким образом перенаправляются к операторам абоненты), будем ориентироваться на результаты имитационного подхода.

**2. РАЗРАБОТКА МОДЕЛИРУЮЩЕГО АЛГОРИТМА: БЛОК-СХЕМА ОБОБЩЕННОГО АЛГОРИТМА ИМИТАЦИИ, БЛОК-СХЕМЫ ЧАСТНЫХ АЛГОРИТМОВ ИМИТАЦИИ.** 

Таким образом, имеем алгоритм, по которому будем строить моделирующую программу.

# **3. РАЗРАБОТКА МОДЕЛИРУЮЩЕЙ ПРОГРАММЫ**

Для разработки моделирующей программы будет использован язык моделирования GPSS (среда GPSS-World), используемый для имитационного моделирования различных систем, в основном систем массового обслуживания. Остановимся поподробней на коде программы.

1. **Создание транзактов:**

GENERATE ,,,3 ; создание трех транзактов-задач

Эта строка генерирует три транзакта (или задачи) сразу. Параметры генерации (в данном случае 3) указывают, что это происходит в одном событии.

1. **Установка приоритета:**

PRIORITY XN1 ; задание приоритета по номеру транзакта (1..3)

Эта команда устанавливает приоритет для текущего транзакта. Приоритет может быть назначен по номеру транзакта, что позволяет системе определять, какой транзакт должен быть обработан первым, если несколько транзактов ожидают обработки.

1. **Инициализация счетчика циклов:**

ASSIGN KOL\_VO,0 ; количество циклов

Здесь переменной KOL\_VO присваивается значение 0. Эта переменная будет использоваться для отслеживания количества циклов обработки для каждого транзакта.

1. **Цикл обработки:**

CYCLE QUEUE Q\_DISPL

Начинается цикл обработки, где транзакт попадает в очередь Q\_DISPL для дисплея.

1. **Занятие дисплея:**

SEIZE DISPL ; занятие дисплея

DEPART Q\_DISPL

ADVANCE 50,30

RELEASE DISPL ; освобождение дисплея

* + SEIZE DISPL — транзакт занимает дисплей.
  + DEPART Q\_DISPL — удаление транзакта из очереди дисплея.
  + ADVANCE 50,30 — выполнение задачи на дисплее, где 50 — среднее время, а 30 — стандартное отклонение.
  + RELEASE DISPL — освобождение дисплея после завершения работы.

1. **Обработка в процессоре:**

QUEUE Q\_CPU ; формирование очереди к процессору

PREEMPT CPU,PR; захват транзактом процессора по приоритету

DEPART Q\_CPU ; освобождение очереди

ADVANCE 100,50 ; работа процессора

RETURN CPU ; снятие прерывания устройства

* + QUEUE Q\_CPU — транзакт попадает в очередь к процессору.
  + PREEMPT CPU, PR — транзакт захватывает процессор с учетом приоритета.
  + DEPART Q\_CPU — удаление транзакта из очереди процессора.
  + ADVANCE 100,50 — выполнение работы на процессоре.
  + RETURN CPU — завершение работы с процессором.

1. **Обработка на принтере:**

QUEUE Q\_PRINT

SEIZE PRINT ; занятие принтера

DEPART Q\_PRINT

ADVANCE 30,10

RELEASE PRINT

* + Транзакт попадает в очередь для принтера.
  + SEIZE PRINT — занятие принтера.
  + DEPART Q\_PRINT — удаление транзакта из очереди принтера.
  + ADVANCE 30,10 — выполнение задачи на принтере.
  + RELEASE PRINT — освобождение принтера.

1. **Обработка анализа:**

QUEUE Q\_ANALIZ ; формирование очереди на анализ

SEIZE ANALIZ

DEPART Q\_ANALIZ

ADVANCE 60,20

RELEASE ANALIZ

* + Транзакт попадает в очередь для анализа.
  + SEIZE ANALIZ — занятие анализатора.
  + DEPART Q\_ANALIZ — удаление транзакта из очереди анализа.
  + ADVANCE 60,20 — выполнение анализа.
  + RELEASE ANALIZ — освобождение анализатора.

1. **Увеличение счетчика циклов:**

ASSIGN KOL\_VO+,1 ; количество циклов +1

Здесь KOL\_VO увеличивается на 1, что означает, что транзакт прошел через один цикл обработки.

1. **Проверка условий завершения:**

TEST E ((PR'E'3)'AND'(P$KOL\_VO'E'100)),1,CYCLE ; проверка если приоритет=3 И количество циклов = 100, то это правда и заканчивается моделирование (ниже), иначе продолжаем в начало

TERMINATE 1

* + TEST E — проверка условий. Если приоритет транзакта равен 3 и количество циклов равно 100, то моделирование завершается.
  + Если условия не выполнены, управление возвращается к началу цикла (CYCLE).

1. **Запуск моделирования:**

START 1

Эта команда начинает выполнение модели.

Таким образом, этот код моделирует процесс обработки транзактов с использованием дисплея, процессора, принтера и анализатора. Он также включает в себя управление приоритетами и отслеживание количества циклов для каждого транзакта, что позволяет завершить моделирование при достижении определенных условий.

Вывод

GPSS World Simulation Report - Untitled Model 4.8.1

Saturday, December 14, 2024 19:01:11

START TIME END TIME BLOCKS FACILITIES STORAGES

0.000 24390.156 26 4 0

NAME VALUE

ANALIZ 10008.000

CPU 10004.000

CYCLE 4.000

DISPL 10002.000

KOL\_VO 10000.000

PRINT 10006.000

Q\_ANALIZ 10007.000

Q\_CPU 10003.000

Q\_DISPL 10001.000

Q\_PRINT 10005.000

LABEL LOC BLOCK TYPE ENTRY COUNT CURRENT COUNT RETRY

1 GENERATE 3 0 0

2 PRIORITY 3 0 0

3 ASSIGN 3 0 0

CYCLE 4 QUEUE 221 0 0

5 SEIZE 221 0 0

6 DEPART 221 0 0

7 ADVANCE 221 0 0

8 RELEASE 221 0 0

9 QUEUE 221 0 0

10 PREEMPT 221 0 0

11 DEPART 221 0 0

12 ADVANCE 221 1 0

13 RETURN 220 0 0

14 QUEUE 220 0 0

15 SEIZE 220 0 0

16 DEPART 220 0 0

17 ADVANCE 220 0 0

18 RELEASE 220 0 0

19 QUEUE 220 0 0

20 SEIZE 220 1 0

21 DEPART 219 0 0

22 ADVANCE 219 0 0

23 RELEASE 219 0 0

24 ASSIGN 219 0 0

25 TEST 219 0 0

26 TERMINATE 1 0 0

FACILITY ENTRIES UTIL. AVE. TIME AVAIL. OWNER PEND INTER RETRY DELAY

DISPL 221 0.441 48.690 1 0 0 0 0 0

CPU 221 0.931 102.693 1 1 0 0 0 0

PRINT 220 0.266 29.476 1 0 0 0 0 0

ANALIZ 220 0.539 59.803 1 2 0 0 0 0

QUEUE MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME AVE.(-0) RETRY

Q\_DISPL 2 0 221 186 0.024 2.641 16.675 0

Q\_CPU 2 0 221 128 0.315 34.805 82.708 0

Q\_PRINT 1 0 220 187 0.020 2.238 14.922 0

Q\_ANALIZ 2 1 220 146 0.085 9.428 28.029 0

CEC XN PRI M1 ASSEM CURRENT NEXT PARAMETER VALUE

2 2 0.000 2 20 21

KOL\_VO 70.000

FEC XN PRI BDT ASSEM CURRENT NEXT PARAMETER VALUE

1 1 24411.466 1 12 13

KOL\_VO 49.000

# **4. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА**

Ниже представлен анализ вывода работы нашей моделирующей программы.

Для анализа результатов работы системы массового обслуживания (СМО) на основе предоставленного отчета GPSS, необходимо выполнить несколько шагов, включая подсчет числа циклов выполнения операций, определение коэффициентов загрузки технических средств и анализ полученных данных.

**1. Подсчет числа циклов выполнения операций**

Согласно вашему отчету, у нас есть следующие операции и их количество:

* **GENERATE:** 3
* **PRIORITY:** 3
* **ASSIGN:** 3
* **CYCLE:** 4
* **SEIZE:** 221
* **DEPART:** 221
* **ADVANCE:** 221
* **RELEASE:** 221
* **QUEUE:** 221
* **PREEMPT:** 221
* **RETURN:** 220
* **TEST:** 219
* **TERMINATE:** 1

Суммируя количество операций, можно определить общее число циклов выполнения:

* **Общее число операций:** 3 + 3 + 3 + 4 + 221 + 221 + 221 + 221 + 221 + 221 + 220 + 219 + 1 = 1, 218

**2. Определение коэффициентов загрузки технических средств**

Коэффициенты загрузки (или утилизации) рассчитываются по формуле:

= ​​

где:

* *T*занятый​ — время, в течение которого ресурс был занят,
* *T*общий​ — общее время моделирования.

**Рассмотрим каждую из технических средств:**

* **DISPL:**
  + Коэффициент загрузки UTIL: 0.441
  + Общее время: 24390.156
  + Время занятости: UDISPL=0.441×24390.156=10751.56
* **CPU:**
  + Коэффициент загрузки UTIL: 0.931
  + Время занятости: UCPU=0.931×24390.156=22645.21
* **PRINT:**
  + Коэффициент загрузки UTIL: 0.266
  + Время занятости: UPRINT=0.266×24390.156=6490.89
* **ANALIZ:**
  + Коэффициент загрузки UTIL: 0.539
  + Время занятости: UANALIZ=0.539×24390.156=13177.63

**3. Анализ результатов**

Теперь, когда мы подсчитали время занятости для каждого ресурса, можно провести анализ:

* **DISPL:**
  + Коэффициент загрузки UTIL 44.1% говорит о том, что дисплей используется менее чем наполовину времени. Это может указывать на возможность оптимизации его использования или уменьшение количества дисплеев.
* **CPU:**
  + Коэффициент загрузки UTIL 93.1% указывает на то, что процессор почти полностью загружен. Это может быть узким местом, и стоит рассмотреть возможность увеличения его мощности или оптимизации процессов, чтобы снизить время ожидания.
* **PRINT:**
  + Коэффициент загрузки UTIL 26.6% также говорит о том, что принтер используется неэффективно. Возможно, стоит рассмотреть возможность уменьшения количества принтеров или перераспределения нагрузки.
* **ANALIZ:**
  + Коэффициент загрузки UTIL 53.9% указывает на то, что анализатор также используется не на полную мощность. Это может быть связано с неравномерным распределением транзактов.

# **5. АНАЛИЗ РЕЗУЛЬТАТОВ**

## **5.1. Моделирование улучшенного варианта системы.**

Чтобы улучшить СМО и значительно снизить коэффициент загрузки ресурсов, можно рассмотреть несколько подходов. Основная цель — уменьшить время обработки и оптимизировать использование ресурсов. Вот несколько предложений для улучшения кода:

**1. Параллелизация обработки**

Если это возможно, можно рассмотреть возможность параллельной обработки транзактов. Это позволит использовать ресурсы более эффективно и сократит время ожидания.

**2. Увеличение числа ресурсов**

Если ресурсы, такие как процессоры или принтеры, являются узким местом, можно увеличить их количество, чтобы уменьшить время ожидания.

**3. Оптимизация алгоритмов обработки**

Проверим, можно ли оптимизировать алгоритмы или изменить последовательность обработки, чтобы улучшить производительность.

GENERATE ,,,3 ; создание трех транзактов-задач

PRIORITY XN1 ; задание приоритета по номеру транзакту (1..3)

ASSIGN KOL\_VO,0 ; количество циклов

CYCLE QUEUE Q\_DISPL

SEIZE DISPL ; занятие дисплея

DEPART Q\_DISPL

ADVANCE 20,10 ; уменьшено время обработки дисплея

RELEASE DISPL ; освобождение дисплея

QUEUE Q\_CPU\_1 ; формирование очереди к первому процессору

QUEUE Q\_CPU\_2 ; формирование очереди ко второму процессору

SEIZE CPU\_1 ; занятие первого процессора

DEPART Q\_CPU\_1 ; освобождение очереди

ADVANCE 40,20 ; уменьшено время обработки процессором

RELEASE CPU\_1 ; освобождение первого процессора

SEIZE CPU\_2 ; занятие второго процессора

DEPART Q\_CPU\_2 ; освобождение очереди

ADVANCE 40,20 ; уменьшено время обработки вторым процессором

RELEASE CPU\_2 ; освобождение второго процессора

QUEUE Q\_PRINT

SEIZE PRINT ; занятие принтера

DEPART Q\_PRINT

ADVANCE 15,5 ; уменьшено время обработки принтера

RELEASE PRINT

QUEUE Q\_ANALIZ ; формирование очереди на анализ

SEIZE ANALIZ

DEPART Q\_ANALIZ

ADVANCE 30,10 ; уменьшено время обработки анализа

RELEASE ANALIZ

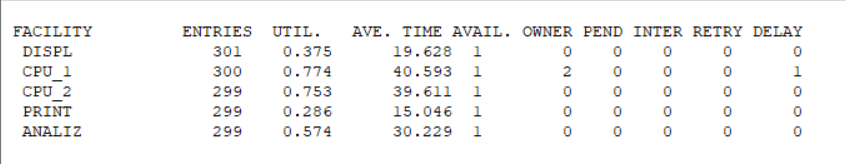
ASSIGN KOL\_VO+,1 ; количество циклов +1

TEST E ((PR'E'3)'AND'(P$KOL\_VO'E'100)),1,CYCLE ; проверка если приоритет=3 И количество циклов = 100, то это правда и заканчивается моделирование (ниже), иначе продолжаем в начало

TERMINATE 1

START 1

## **5.2. Интерпретация результатов и рекомендации по реализации моделируемой системы.**



Для анализа изменений коэффициента загрузки (UTIL) в первоначальной и новой системах, мы можем сравнить значения UTIL для каждого компонента.

Первоначальная система:

* **DISPL**: 0.441
* **CPU**: 0.931
* **PRINT**: 0.266
* **ANALIZ**: 0.539

Новая система:

* **DISPL**: 0.375
* **CPU\_1**: 0.774
* **CPU\_2**: 0.753
* **PRINT**: 0.286
* **ANALIZ**: 0.574

**Изменения по каждому компоненту:**

1. **DISPL**:
   * Первоначальная: 0.441
   * Новая: 0.375
   * Изменение: -0.066 (уменьшение)
2. **CPU**:
   * Среднее значение для CPU в новой системе: (0.774 + 0.753) / 2 = 0.7635
   * Первоначальная: 0.931
   * Изменение: 0.7635 - 0.931 = -0.1675 (уменьшение)
3. **PRINT**:
   * Первоначальная: 0.266
   * Новая: 0.286
   * Изменение: +0.020 (увеличение)
4. **ANALIZ**:
   * Первоначальная: 0.539
   * Новая: 0.574
   * Изменение: +0.035 (увеличение)

**Интерпретация результатов:**

1. **Эффективность использования ресурсов**: Уменьшение загрузки для DISPL и CPU свидетельствует о более эффективном использовании этих ресурсов в новой системе. Однако увеличение UTIL для PRINT и ANALIZ может указывать на необходимость дополнительного внимания к этим компонентам.
2. **Разделение CPU**: Введение двух процессоров (CPU\_1 и CPU\_2) позволило снизить среднюю загрузку на каждый из них, но в то же время общий коэффициент загрузки увеличился. Это может быть связано с увеличением общего числа транзактов, обрабатываемых системой.

**Рекомендации по реализации моделируемой системы:**

1. **Мониторинг и анализ**: необходимо регулярно мониторить загрузку всех компонентов системы, чтобы выявлять узкие места и оптимизировать производительность.
2. **Оптимизация процессов**: рассмотреть возможность оптимизации процессов, связанных с PRINT и ANALIZ, чтобы уменьшить их загрузку. Это может включать в себя улучшение алгоритмов обработки данных или увеличение числа ресурсов.
3. **Параллелизация задач**: если это возможно, продолжить использовать параллелизацию для обработки задач, особенно для CPU, чтобы улучшить производительность и снизить время ожидания.
4. **Балансировка нагрузки**: рассмотреть возможность балансировки нагрузки между CPU\_1 и CPU\_2, чтобы избежать перегрузки одного из процессоров.
5. **Тестирование и итерации**: провести тестирование системы с различными сценариями нагрузки и производительности, чтобы определить оптимальные параметры для достижения наилучших результатов.

Эти рекомендации помогут улучшить эффективность и производительность моделируемой системы, а также снизить общий коэффициент загрузки, что приведет к более быстрому и надежному обслуживанию транзактов.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Советов, Б. Я.Моделирование систем. Практикум [Текст] : учеб. пособие / Б.Я. Советов, С.А. Яковлев. - 2-е изд., перераб. и доп. - М. : Высш. шк., 2003. - 295 с. : ил.
2. Знакомство с основными операторами GPSS World<https://www.matematicus.ru/gpss-world/znakomstvo-s-osnovnymi-operatorami-gpss-world-urok-2>
3. Математическое моделирование систем связи : учебное пособие /К. К. Васильев, М. Н. Служивый. – Ульяновск : УлГТУ, 2008 – 170 с.
4. GPSS World основы имитационного моделирования различных систем [Kudryavtsev-E.M.-GPSS-World.-Osnovy-imitatsionnogo-modelirovaniya-razlichny-h-sistem-2004.pdf - Яндекс Документы](https://docs.yandex.ru/docs/view?tm=1734197079&tld=ru&lang=ru&name=Kudryavtsev-E.M.-GPSS-World.-Osnovy-imitatsionnogo-modelirovaniya-razlichny-h-sistem-2004.pdf&text=gpss%20world&url=http%3A%2F%2Flazysmart.ru%2Fwp-content%2Fuploads%2F2016%2F08%2FKudryavtsev-E.M.-GPSS-World.-Osnovy-imitatsionnogo-modelirovaniya-razlichny-h-sistem-2004.pdf&lr=10865&mime=pdf&l10n=ru&sign=52c3e7cf9012eb6eceff7fde4930f093&keyno=0&nosw=1&serpParams=tm%3D1734197079%26tld%3Dru%26lang%3Dru%26name%3DKudryavtsev-E.M.-GPSS-World.-Osnovy-imitatsionnogo-modelirovaniya-razlichny-h-sistem-2004.pdf%26text%3Dgpss%2Bworld%26url%3Dhttp%253A%2F%2Flazysmart.ru%2Fwp-content%2Fuploads%2F2016%2F08%2FKudryavtsev-E.M.-GPSS-World.-Osnovy-imitatsionnogo-modelirovaniya-razlichny-h-sistem-2004.pdf%26lr%3D10865%26mime%3Dpdf%26l10n%3Dru%26sign%3D52c3e7cf9012eb6eceff7fde4930f093%26keyno%3D0%26nosw%3D1)

# **ПРИЛОЖЕНИЕ А**

**Код моделируемой системы в GPSS World**

GENERATE ,,,3 ;создание трех транзактов-задач

PRIORITY XN1 ;задание приортета по номеру транзакту (1..3)

ASSIGN KOL\_VO,0 ;количество циклов

CYCLE QUEUE Q\_DISPL

SEIZE DISPL ;занятие дисплея

DEPART Q\_DISPL

ADVANCE 50,30

RELEASE DISPL ; освобождение дисплея

QUEUE Q\_CPU ;формирование очереди к процессору

PREEMPT CPU,PR;захват транзактом процессора по приоритету

DEPART Q\_CPU ;освобождение очереди

ADVANCE 100,50 ;работа процессора

RETURN CPU ;снятие прерывания устройства

QUEUE Q\_PRINT

SEIZE PRINT ;занятие принтера

DEPART Q\_PRINT

ADVANCE 30,10

RELEASE PRINT

QUEUE Q\_ANALIZ ;формирование очереди на анализ

SEIZE ANALIZ

DEPART Q\_ANALIZ

ADVANCE 60,20

RELEASE ANALIZ

ASSIGN KOL\_VO+,1 ; количество циклов +1

TEST E ((PR'E'3)'AND'(P$KOL\_VO'E'100)),1,CYCLE ;проверка если приоритет=3 И колиество циклов = 100, то это правда и заканчивается моделирование (ниже), иначе продолжаем в начало

TERMINATE 1

START 1