Задания для самостоятельной работы. GPSS

Отчёт по лабораторной работе №17

Ибатулина Дарья Эдуардовна

Содержание

# 1 Цель работы

Реализовать с помощью gpss задания для самостоятельной работы.

# 2 Задание

Реализовать с помощью gpss:

* модель работы вычислительного центра;
* модель работы аэропорта;
* модель работы морского порта.

# 3 Теоретическое введение

GPSS (General Purpose Simulation System) — это один из первых специализированных языков программирования для имитационного моделирования, созданный в 1961 году американским инженером Джеффри Гордоном в корпорации IBM. Первоначально язык разрабатывался для нужд моделирования сложных логистических и производственных процессов в промышленных и военных системах, где требовался учёт случайных событий и взаимодействия большого количества объектов во времени.

GPSS стал знаковым инструментом в истории моделирования: он заложил основы событийного подхода и ввёл понятие транзакта как активного объекта, перемещающегося по блокам логики системы. Эти концепции впоследствии легли в основу многих других языков и программных сред моделирования. Благодаря модульной структуре и простой записи моделей, GPSS получил широкое распространение в университетах и научных учреждениях как средство обучения и анализа дискретных систем.

Практическое применение GPSS охватывает широкий спектр задач:

* Организация работы производственных цехов: моделирование потока деталей между станками, учёт времени обработки, простоев и загрузки оборудования;
* Системы массового обслуживания: моделирование очередей в банках, поликлиниках, аэропортах с целью оценки времени ожидания и необходимости в дополнительном персонале;
* Логистика и склады: моделирование перемещения товаров между зонами хранения, погрузки и разгрузки, анализ загрузки транспортных средств;
* Транспорт: моделирование движения автобусов, поездов, планирование расписаний с учётом времени на посадку и высадку пассажиров;
* Военные приложения: планирование операций снабжения, имитация действий в сложных логистических цепочках.

Одним из достоинств GPSS является то, что язык допускает использование случайных величин (например, времени обслуживания или интервалов между заявками), что позволяет создавать реалистичные модели, приближенные к поведению реальных систем. Также GPSS даёт возможность легко собирать статистику по ключевым метрикам: времени пребывания объектов в системе, загрузке ресурсов, количеству отказов и пр.

Несмотря на то, что с момента своего создания прошло более шестидесяти лет, GPSS продолжает использоваться как в учебных целях, так и в инженерной практике благодаря своей простоте, наглядности и эффективности в решении прикладных задач, связанных с анализом и оптимизацией дискретных процессов.

При выполнении работы были использованы источники [1,2].

# 4 Выполнение лабораторной работы

## 4.1 Модель работы вычислительного центра

На вычислительном центре в обработку принимаются три класса заданий А, В и С. Исходя из наличия оперативной памяти ЭВМ задания классов А и В могут решаться одновременно, а задания класса С монополизируют ЭВМ.

Задания класса А поступают через мин, класса В — через мин, класса С — через мин.

Требуют для выполнения: класс А — мин, класс В — 21 мин, класс С — мин.

Задачи класса С загружаются в ЭВМ, если она полностью свободна. Задачи классов А и В могут дозагружаться к решающей задаче. Необходимо смоделировать работу ЭВМ за 80 ч. Определить её загрузку.

Я использовала при моделировании многоканальное устройство ram на две заявки. Классы A, B занимают по одному месту (поскольку могут выполняться одновременно), а класс С занимает сразу два места. Таймер выставила на 4800 минут (80\*60). В итоге я получила следующую модель (рис. [1](#fig:001)).

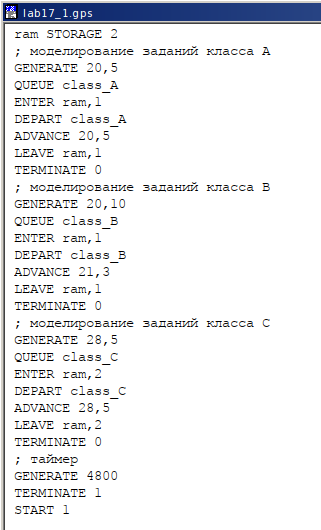


Figure 1: Модель работы вычислительного центра

После запуска симуляции я получила отчёт (рис. [2](#fig:002)).

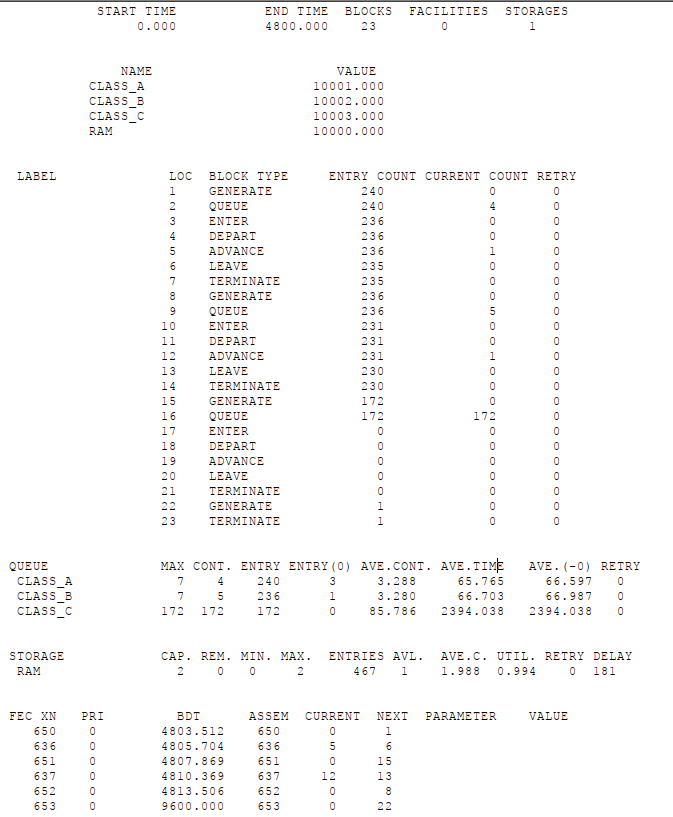


Figure 2: Отчёт по модели работы вычислительного центра

В задании требовалось определить загрузку ЭВМ - она равна 0.994.

## 4.2 Модель работы аэропорта

Самолёты прибывают для посадки в район аэропорта каждые мин. Если взлетно-посадочная полоса свободна, прибывший самолёт получает разрешение на посадку. Если полоса занята, самолет выполняет полет по кругу и возвращается в аэропорт каждые 5 мин. Если после пятого круга самолет не получает разрешения на посадку, он отправляется на запасной аэродром.

В аэропорту через каждые мин к взлетно -посадочной полосе выруливают готовые к взлёту самолёты и получают разрешение на взлёт, если полоса свободна. Для взлета и посадки самолёты занимают полосу ровно на 2 мин. Если при свободной полосе одновременно один самолёт прибывает для посадки, а другой - для взлёта, то полоса предоставляется взлетающей машине.

Требуется:

* выполнить моделирование работы аэропорта в течение суток;
* подсчитать количество самолётов, которые взлетели, сели и были направлены на запасной аэродром;
* определить коэффициент загрузки взлетно-посадочной полосы.

Модель получилась следующая (рис. [3](#fig:003)):

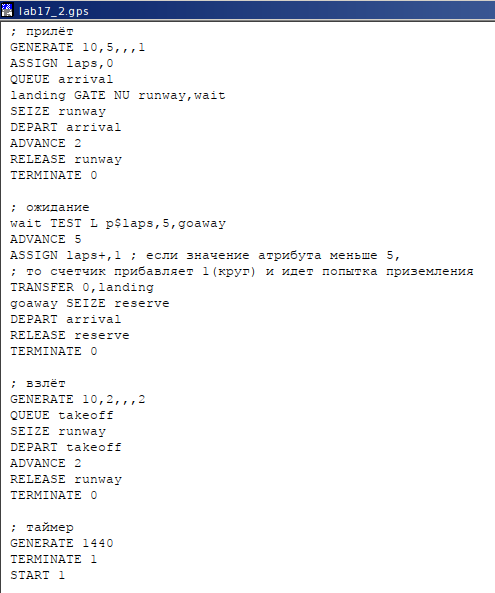


Figure 3: Модель работы аэропорта

После запуска симуляции я получила отчёт (рис. [4](#fig:004)).

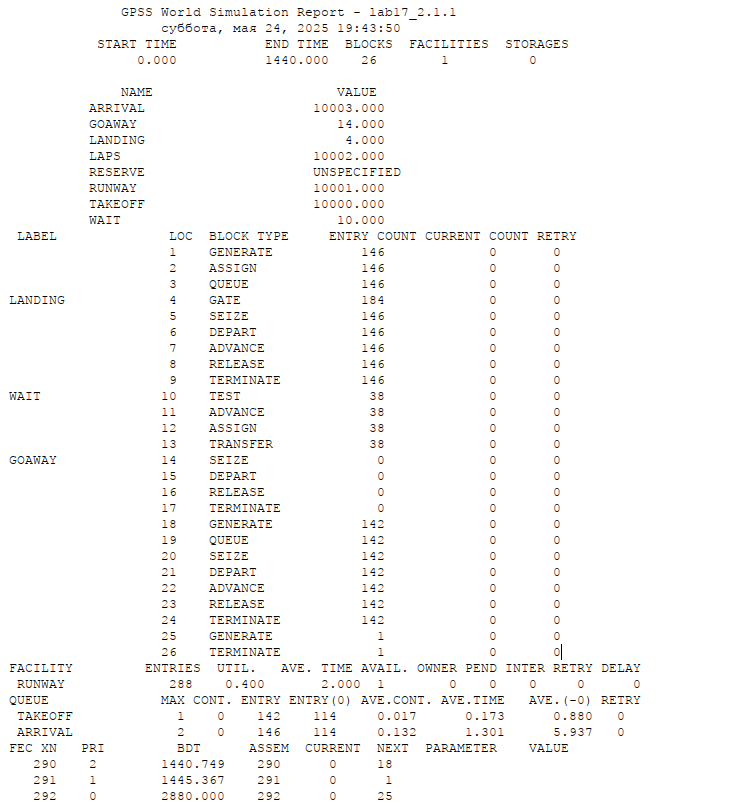


Figure 4: Отчёт по модели работы аэропорта

Прибытие самолётов задаётся командой GENERATE 10,5,,,1. Для каждого нового самолёта устанавливается счётчик кругов: ASSIGN laps,0 - установка атрибута laps (счётчик кругов) в 0 для нового самолёта.

Далее самолёт пытается занять посадочную полосу. Если полоса занята, самолёт переходит к ожиданию (метка wait). Здесь проверяется, сколько кругов уже сделал самолёт: TEST L p$laps,5,goaway – если значение атрибута p$laps < 5, переход к goaway. Если кругов меньше пяти, самолёт ждёт 5 минут, делая круг над аэродромом: ADVANCE 5 и увеличивает свой счётчик: ASSIGN laps+1,1. После этого самолёт снова пытается занять полосу.

Если же самолёт сделал уже 5 кругов, он уходит на запасную полосу (метка other): SEIZE reserve.

Также в коде реализован процесс взлёта: GENERATE 10,2,,,2 создаёт поток самолётов на взлёт, которые занимают полосу (SEIZE runway), взлетают (ADVANCE 2) и освобождают её (RELEASE runway).

В конце моделирования используется таймер: GENERATE 1440 и TERMINATE 1, что завершает моделирование через 1440 минут (сутки).

Взлетело 142 самолета, село 146, а в запасной аэропорт отправилось 0. В запасной аэропорт не отправились самолеты, поскольку процессы обработки длятся всего 2 минуты, что намного быстрее, чем генерации новых самолетов (время между пребыванием новых самолётов в аэропорт может быть равно от 5 до 15 минут, а между взлётами самолётов - от 8 до 12 минут). Коэффициент загрузки полосы равняется 0.400, что означает, что полоса большую часть времени не используется.

## 4.3 Модель работы морского порта

Морские суда прибывают в порт каждые часов. В порту имеется N причалов. Каждый корабль по длине занимает M причалов и находится в порту часов. Требуется построить GPSS-модель для анализа работы морского порта в течение полугода, определить оптимальное количество причалов для эффективной работы порта.

Рассмотрим два варианта исходных данных:

1. ч, ч, ч, ч, , ;
2. ч, ч, ч, ч, , .

**Первый вариант модели**

В этой модели я снова использовала многоканальное устройство и через него задавала количество причалов. После этого всё шло по стандартной схеме - “генерировалось” судно, оно вставало в очередь, после - получало место у причалов, занимая три причала. Заняв место, судно покидало очередь, проводило необходимое время у причала. Потом освобождаются занятые места (3 причала), так как судно отчаливает.

Модель получилась следующая (рис. [5](#fig:005)):

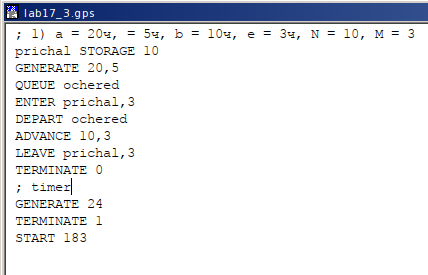


Figure 5: Модель работы морского порта (1 вариант)

После запуска симуляции я получила отчёт (рис. [6](#fig:006)).

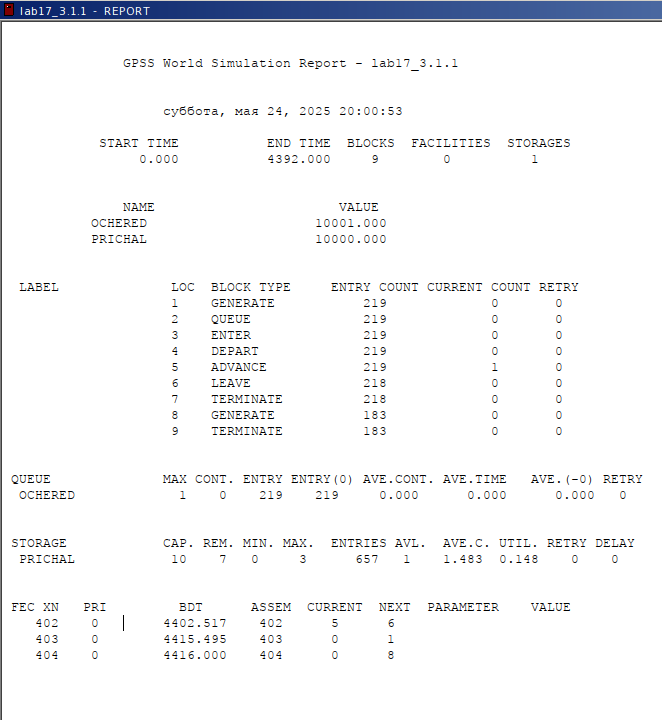


Figure 6: Отчёт по модели работы морского порта (1 вариант)

Судна обрабатываются быстрее, чем успевают приходить новые, так как очередь не набирается. По отчёту видно, что коэффициент загрузки довольно мал - всего 0.148. Это указывает на то, что подбирая оптимальное количество причалов, их можно уменьшить. Наименьшее возможное число причалов равно 3, так как каждый корабль в длину занимает именно 3 причала. Я заменила строчку prichal STORAGE 10 на prichal STORAGE 3 и получила следующее (рис. [7](#fig:007)):

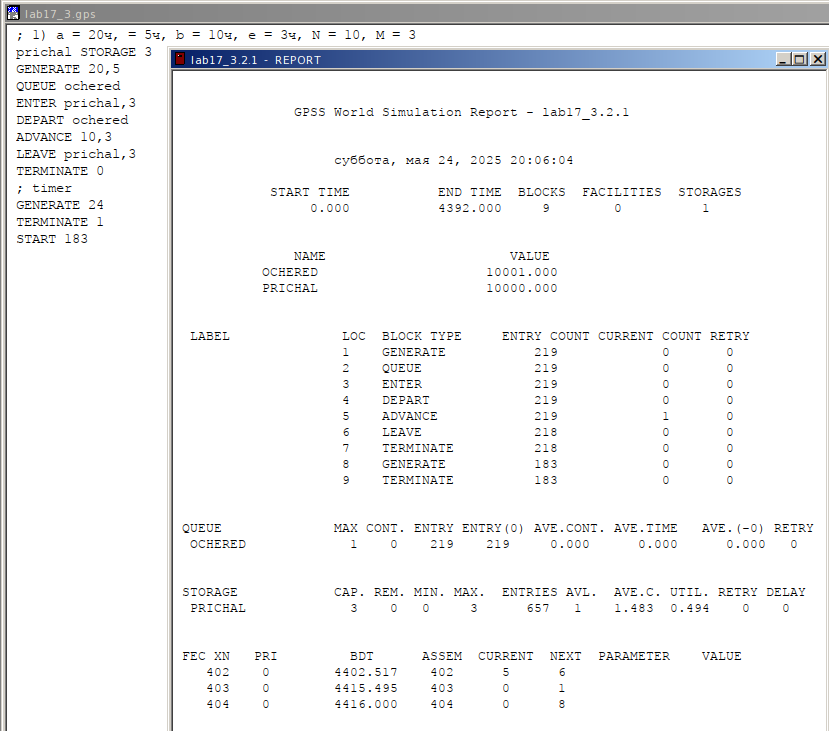


Figure 7: Оптимизированная модель работы морского порта (1 вариант) и отчёт

Здесь уже коэффициент загрузки равен 0.495, что является более эффективным.

**Второй вариант модели**

Во втором варианте модели у нас изменились лишь параметры, поэтому модель выглядит идентично (рис. [8](#fig:008), [9](#fig:009)):

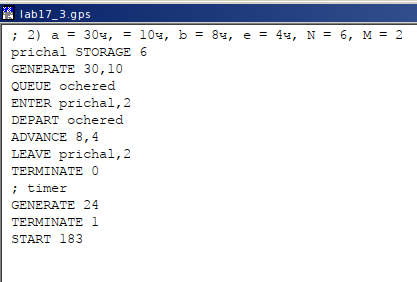


Figure 8: Модель работы морского порта (2 вариант)

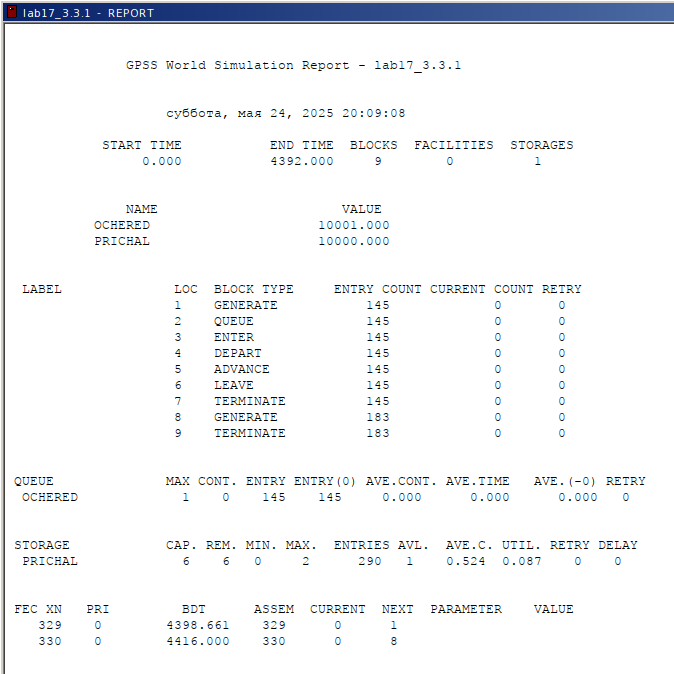


Figure 9: Отчёт по модели работы морского порта (2 вариант)

В задании также сказано оптимизировать модель, потому что на данном этапе коэффициент загрузки крайне мал - всего 0.087. Так что, как и в прошлый раз, я сократила количество причалов до минимально возможного - 2 в данном случае (рис. [10](#fig:010)).

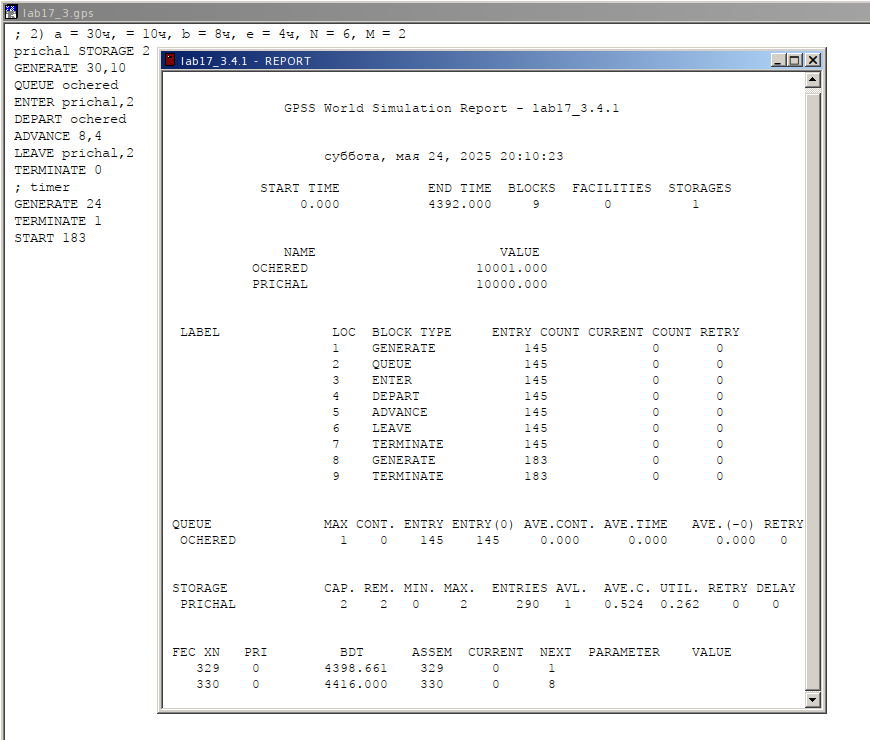


Figure 10: Оптимизированная модель работы морского порта (2 вариант) и отчёт

Видно, что коэффициент загрузки стал равен 0.262, что уже выше, чем в прошлый раз.

# 5 Выводы

В результате выполнения данной лабораторной работы я реализовала с помощью gpss:

* модель работы вычислительного центра;
* модель работы аэропорта;
* модель работы морского порта.

# Список литературы

1. GPSS-WORLD, основы имитационного моделирования на живых примерах [Электронный ресурс]. URL: <https://habr.com/ru/articles/192044/>.

2. М. К.Е. GPSS World. Основы имитационного моделирования различных систем. Москва: ДМК Пресс, 2004. 318 с.