

Лабораторная работа №1

Введение

Выполнение каждой лабораторной работы складывается из нескольких стадий: подготовка, разработка программы, получение результатов, оформление и сдача отчетов по работе.

Оформление и сдача отчета по лабораторной работе. Каждую лабораторную работу студент оформляет в виде отчета, который должен содержать:

- исходные данные по вариантам;
- структурную схему моделируемой системы;
- принятую единицу модельного времени;
- таблицу определений

Элементы GPSS	Интерпретация
Транзакты: Приборы: Многоканальные устройства: Очереди: Функции:	

- текст программы;
- полученные результаты в форме таблиц;
- необходимые расчеты и выводы по работе

Пакет GPSS (General Purpose Simulating System - общецелевая система моделирования) является одним из наиболее эффективных и распространенных программных средств моделирования дискретных систем на IBM PC и успешно используется для моделирования систем, формализуемых в виде схем массового обслуживания (СМО).

Примерами таких схем могут служить: телефонные станции, ремонтные мастерские, билетные кассы, справочные бюро, парикмахерские и т.п. Каждая такая система состоит из определенного числа обслуживающих единиц, называемых «каналами» обслуживания. В качестве каналов могут фигурировать: линии связи; лица, выполняющие те или иные операции;

различные приборы и т.д. Системы массового обслуживания могут быть как одно-, так и многолокальными.

Работа любой системы массового обслуживания состоит в выполнении поступающего в нее потока требований или заявок. Заявки поступают одна за другой в некоторые, вообще говоря, случайные моменты времени. Обслуживание поступившей заявки продолжается какое-то время, после чего канал освобождается и готов для приема следующей заявки.

Случайный характер потока заявок (требований) и времени обслуживания приводят к тому, что в какие-то периоды времени на входе СМО скапливается излишнее количество заявок (они либо становятся в очередь, либо покидают СМО необслуженными); в другие же периоды СМО будет работать с недогрузкой или вообще простаивать.

Работа СМО представляет собой случайный процесс с дискретными состояниями и непрерывным временем; состояние СМО меняется скачком в моменты появления каких-то событий (или прихода новой заявки, или окончания обслуживания, или момента, когда заявка, которой надоело ждать, покидает очередь).

С помощью имитационного моделирования на GPSS возможно построение моделей, связывающих заданные условия работы СМО (число каналов, их производительность, правила работы, характер потока заявок) с интересующими нас характеристиками показателями эффективности СМО, описывающими, с той или другой точки зрения, ее способность справляться с потоком заявок. В качестве таких показателей (в зависимости от обстановки и целей исследования) могут применяться, например, среднее число заявок, обслуженных СМО в единицу времени; среднее число занятых каналов; среднее число заявок в очереди и среднее время обслуживания и т.д.

СМО делятся на типы (классы) по ряду признаков: первое деление - СМО с отказами и СМО с очередью.

В СМО с отказами заявка, поступившая в момент, когда все каналы заняты, получает отказ, покидает СМО и в дальнейшем процессе

обслуживания не участвует. Примеры СМО с отказами встречаются в телефонии: заявка на разговор, пришедшая в момент, когда все каналы связи заняты, получает отказ и покидает СМО необслуженной.

В СМО с очередью заявка, пришедшая в момент, когда все каналы заняты, не уходит, а становится в очередь и ожидает возможности быть обслуженной. Этот вариант встречается чаще всего, недаром теория массового обслуживания имеет второе название - «теория очередей».

СМО с очередью подразделяются на разные виды в зависимости от того, как организована очередь, - ограничена она или не ограничена. Ограничения могут касаться как длины очереди, так и времени ожидания.

При анализе СМО должна учитываться также и «дисциплина обслуживания»: заявки могут обслуживаться либо в порядке поступления (раньше пришла - раньше обслуживается), либо в случайном порядке.

Нередко встречается так называемое обслуживание с приоритетом - некоторые заявки обслуживаются вне очереди.

Существуют и другие классы СМО, которые подробно рассматриваются в книгах по теории вероятностей.

Модель СМО может быть представлена в виде блок-схемы, которая представляет собой набор фигур с характерным описанием блоков, соединенных между собой линиями. Вид каждого блока стандартен.

Понятие транзакта. Конфигурация блок-схемы GPSS-модели отражает направления, по которым происходит движение перемещающихся элементов. Каждый такой элемент называется транзактом.

Транзакт – динамический (движущийся) элемент GPSS-модели (например: покупатель, автомобиль на мойке, заявка на складе).

Работа GPSS-модели заключается в перемещении транзактов от блоков к блокам.

В самом начале моделирования в GPSS-модели нет ни одного транзакта. В процессе моделирования транзакты входят в модель в определенные

моменты в соответствии с логическими потребностями, которые возникают в моделируемой системе.

Подобным же образом транзакты покидают модель в определенные моменты в зависимости от специфики моделирования.

В общем случае в модели существует большое количество транзактов, однако в один момент времени движется только один транзакт.

Если транзакт начал движение, он перемещается от блока к блоку, вызывая выполнение соответствующих подпрограмм.

Продвижение транзакта продолжается до тех пор, пока не произойдет одно из следующих возможных событий:

1. Транзакт входит в блок, функция которого - задержка транзакта на некоторое определенное модельное время (например, кресло парикмахера).
2. Транзакт входит в блок, который удаляет транзакт из модели.
3. Транзакт пытается войти в некоторый блок согласно блок-схеме, но блок отказывается его принять. В этом случае транзакт остается в том блоке, в котором он находится. Позднее он повторит свою попытку войти в блок. Одна из таких попыток может оказаться успешной, после чего транзакт может продолжать свое перемещение в модели.

Если возникло одно из описанных условий, транзакт остается на месте, и начинается перемещение в модели другого транзакта.

В общем случае выполнение модели заключается в последовательном обращении к подпрограммам, являющемся следствием входа в определенные блоки перемещающихся транзактов.

Различные события реальных систем происходят в течение некоторого периода времени. Клиенты приходят в парикмахерскую; когда приходит их очередь, они попадают на обслуживание к парикмахеру. Стрижка заканчивается, и клиент покидает парикмахерскую. Если все эти события представить в модели, то их возникновение должно происходить на фоне модельного времени.

Таймер модельного времени корректируется автоматически, в соответствии с логикой модельного процесса.

Особенности таймера:

1. Таймер GPSS регистрирует только целые значения. Это означает, что события могут возникать только в «целые» моменты времени.

2. Единицу времени, которая может быть отмечена таймером, определяет разработчик. Разработчик следит за тем, чтобы все данные, связанные со временем, были выражены через эту определенную единицу времени.

Различие между модельным и реальным временем. Когда значение таймера модельного времени продвигается в следующую точку, происходит некоторая задержка, чтобы скорректировать модель. В реальной же системе такого времени на корректировку не существует.

Не исключено, что для моделирования только одной минуты модельного времени потребуется несколько десятков минут реального времени (например, при моделировании вычислительных систем).

С другой стороны, эксперименты, которые проводят в течение нескольких недель, могут занимать всего несколько секунд реального времени при моделировании на ЭВМ.

Выходная статистика

Отформатированный файл статистики состоит из подразделов, содержащих стандартную статистику об объектах GPSS\PC, используемых в данной модели (FACILITY, QUEUE, STORAGE и т.д.). Начинается файл статистики с заголовка, который берется из поля комментария, расположенного перед началом программы. Заголовок появляется на каждой странице файла статистики. После заголовка автоматически вставляется подзаголовок, который содержит имя неотформатированного файла статистики, номер версии GPSS\PC, серийный номер, дату и время моделирования.

Пример: GPSS\PC Report file TEST (V2) 06-24-1989 21:57:38

Далее следует строка, содержащая основную информацию о результатах работы модели. Например:

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES	FREE MEMORY
0	10850	24	1	1	15850

Элементы статистики, представленные в этой строке имеют следующее содержание:

- START TIME – абсолютное системное время в момент начала моделирования. Оно эквивалентно абсолютному системному времени после последнего применения операторов RESET или CLEAR;

- END TIME – абсолютное время, когда счетчик завершений принимает значение 0;

- BLOCKS – количество блоков, использованных в текущей модели, к моменту завершения моделирования;

- FACILITIES – количество устройств, использованных в модели, к моменту завершения моделирования;

- STORAGES – количество многоканальных устройств, использованных в текущей модели, к моменту завершения моделирования;

- FREE MEMORY – количество байтов памяти, доступной для дальнейшего использования.

Затем в файле статистики следует информация об именах, которые просматривает GPSS/PC в ходе моделирования. Информация об именах имеет следующий вид:

NAME	VALUE	TYPE
MOTOR	10001	2

Поле NAME отмечает имена, содержащиеся в программе модели.

Поле VALUE определяет числовое значение, соответствующее имени. Система устанавливает начальный номер равным 10000.

Поле TYPE равно 0, если значение имени устанавливает пользователь; равно 2, если значение имени устанавливает система; 3 – если имя является именем блока.

Далее описываются блоки текущей модели в виде:

LINE	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
90	1	GENERATE	333	0	0

Поле LINE определяет номер строки в рабочей модели, связанный с блоком GPSS/PC.

Поле LOC определяет имя или номер этого блока.

Поле BLOCK TYPE определяет тип блока GPSS/PC.

Поле ENTRY COUNT определяет количество транзактов, вошедших в данный блок, после последнего выполнения блоков RESET или CLEAR, или с начала работы программы.

Поле CURRENT COUNT определяет количество транзактов, находящихся в данном блоке в конце моделирования.

Поле RETRY определяет количество транзактов, ожидающих специальных условий, зависящих от состояния данного блока.

Если в модели используются объекты типа "устройство", то далее в файле статистики идет информация об этих объектах. Например:

FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAILABLE	OWNER	PEND	INTER
TELLER	254	0.996	395.67	1	291	0	0
RETRY	DELAY						
0	78						

Поле FACILITY определяет номер или имя объекта типа "устройство".

Поле ENTRIES определяет количество раз, когда устройство было занято или прервано после последнего выполнения блоков RESET или CLEAR или с начала работы программы.

Поле UTIL. определяет часть периода моделирования, в течение которого устройство было занято.

Поле AVE.TIME определяет среднее время занятости устройства одним сообщением в течение периода моделирования после последнего выполнения операторов RESET или CLEAR.

Поле AVAILABLE определяет состояние готовности устройства в конце периода моделирования. Оно равно 1, если устройство готово и 0 – если не готово.

Поле OWNER определяет номер последнего сообщения, занимавшего устройство. 0 означает, что устройство не занималось.

Поле PEND определяет количество сообщений, ожидающих устройство, находящееся в "режиме прерывания".

Поле INTER определяет количество сообщений, прерывающих устройство в данный момент (счетчик сообщений в списке прерывания).

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния объекта типа "устройство".

Поле DELAY определяет количество сообщений, ожидающих занятия устройства. Сюда входят также сообщения, ожидающие освобождения устройства в "режиме прерывания" (блок PREEMPT).

В случае использования в модели объектов типа "очередь", далее следует информация об этих объектах. Например:

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
MOD	78	78	332	1	36.24	11613.51	11848.6	0

Поле QUEUE определяет имя или номер объекта типа "очередь".

Поле MAX определяет максимальное содержимое объекта типа "очередь" в течение периода моделирования, который начинается с начала работы программы или с последнего оператора RESET или CLEAR.

Поле CONT определяет текущее содержимое объекта типа "очередь" в конце периода моделирования.

Поле ENTRIES определяет общее количество входов в очередь в течение периода моделирования (счетчик входов).

Поле ENTRIES(0) определяет общее количество входов в очередь с нулевым временем ожидания (счетчик "нулевых" входов).

Поле AVE.CONT определяет среднее значение содержимого очереди.

Поле AVE.TIME определяет среднее время, проведенное в очереди с учетом всех входов в очередь.

Поле AVE.(-0) определяет среднее время, проведенное в очереди без учета "нулевых" входов в очередь.

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния объекта типа "очередь".

Если в модели использовались объекты типа "многоканальное устройство", то далее в файле статистики идет информация об этих объектах. Например:

STORAGE	CAP.	REMAIN	MIN	MAX	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
MOTOR	3	3	0	1	50	1	0.99	0.331	0	0

Поле STORAGE определяет имя или номер объекта типа "многоканальное устройство".

Поле CAP. определяет емкость многоканального устройства, заданную оператором STORAGE.

Поле REMAIN определяет число единиц свободной емкости многоканального устройства в конце периода моделирования.

Поле MIN определяет минимальное количество используемой емкости многоканального устройства за период моделирования.

Поле MAX определяет максимальное количество используемой емкости многоканального устройства за период моделирования.

Поле ENTRIES определяет количество входов в многоканальное устройство за период моделирования.

Поле AVL. определяет состояние готовности многоканального устройства в конце периода моделирования. 1 означает, что многоканальное устройство готово, 0 – не готово.

Поле AVE.C определяет среднее значение занятой емкости за период моделирования.

Поле UTIL. определяет часть периода моделирования, в течение которого многоканальное устройство использовалось.

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния многоканального устройства.

Поле DELAY определяет количество сообщений, ожидающих возможности входа в блок ENTER.

Далее в файле статистики выводятся списки пользователя. Например:

USER CHAIN	CHAIN SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
TAXILINE	202	0	100.70	252	203	40289.50

Поле USER CHAIN определяет номер или имя объекта типа "список пользователя".

Поле CHAIN SIZE определяет количество сообщений в списке пользователя в конце периода моделирования.

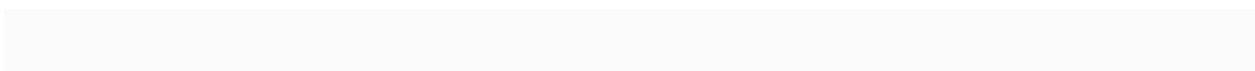
Поле RETRY определяет количество сообщений, ожидающих наступления специальных условий, связанных с состоянием объекта типа "список пользователя".

Поле AVE.CONT определяет среднее содержимое списка пользователя в течение периода моделирования.

Поле ENTRIES определяет общее количество сообщений, помещаемых в список пользователя в течение периода моделирования.

Поле MAX определяет максимальное количество транзактов в списке пользователя за период моделирования.

Поле AVE.TIME определяет среднее время пребывания транзакта в списке пользователя.



Моделирование системы с прибором

Внесение транзактов в модель

Блок GENERATE (ГЕНЕРИРОВАТЬ).

GENERATE - это блок, через который транзакты входят в модель.

В одной модели может быть любое число блоков GENERATE.

Изображение:



Операнды:

A - средний интервал времени (по умолчанию - ноль);

B - половина поля допуска равномерно распределенного интервала (по умолчанию - ноль);

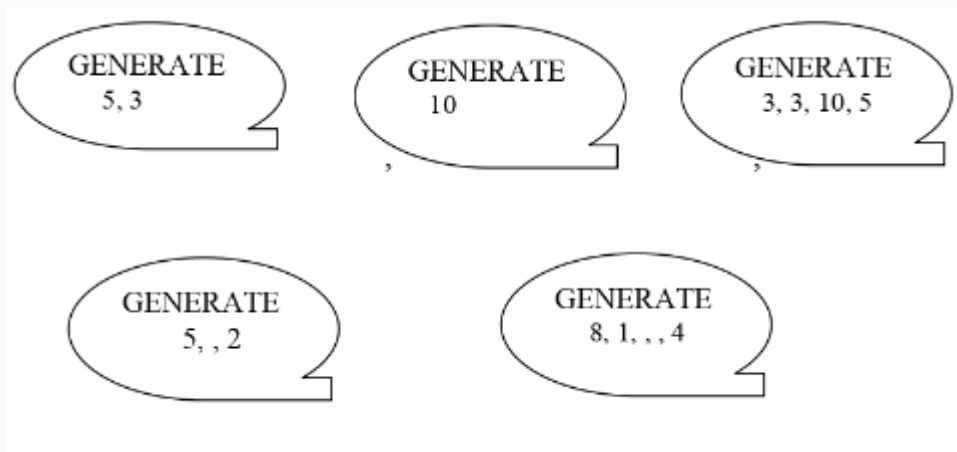
C - смещение интервала (по умолчанию смещение отсутствует);

D - ограничитель количества транзактов, проходящих через блок (по умолчанию – условно бесконечно);

E - уровень приоритета (по умолчанию - 0).

Любой из операндов может отсутствовать.

Пример:



Удаление транзактов из модели

Блок TERMINATE (ЗАВЕРШИТЬ). Транзакты удаляются из модели, попадая в блок TERMINATE. Изображение блока:



Операнд А - указатель уменьшения счетчика завершений (по умолчанию - ноль).

В модели может быть несколько блоков TERMINATE, но счетчик завершений только один. Счетчик завершений - ячейка памяти ЭВМ, которая хранит некоторое значение, записанное в начале моделирования. В процессе моделирования транзакты попадают в блоки TERMINATE, и таким образом происходит уменьшение содержания счетчика на параметр А (обычно $A=1$, но не обязательно).

Обычно счетчик завершений задается параметром А команды начала моделирования START (о командах управления процессом моделирования см. ниже).

Элементы, символизирующие обслуживающие приборы

Элементами, представляющими прибор, может быть либо человек, либо какой-то предмет. Люди: парикмахер, заправщик бензоколонки, продавец. Предметы: стоянка автомобилей, подъемный кран и др.

Приборы характеризуются двумя основными свойствами:

1. Каждый прибор в любой момент времени обслуживает только одно требование. Если в процессе обслуживания появляется новое требование, оно либо ждет, либо отправляется в другое место системы, либо прерывает протекающее обслуживание до его завершения.

2. Когда на прибор поступает требование, оно задерживается в блоке прибора на некоторое время, называемое временем обслуживания.

Поскольку в различных точках системы может быть несколько приборов, то, чтобы различать их, им могут быть давать имена. Числовые - положительное целое; символические - 3-5 алфавитно- цифровых символа, причем первые три - буквы.

Занятие свободных приборов

Блоки SEIZE (ЗАНЯТЬ) и RELEASE (ОСВОБОДИТЬ)

Предположим, что мы хотим использовать прибор. При этом мы должны выполнить следующие шаги:

1. Ожидаем своей очереди, если это необходимо. Естественно, что ожидание требует некоторого интервала времени.
2. Когда приходит наша очередь, занимаем прибор. Событие «занятие прибора» происходит в некоторой точке времени.
3. Прибор остается в состоянии занятости, пока не закончится обслуживание. Для обслуживания требуется также некоторый интервал времени.
4. Когда обслуживание заканчивается, мы освобождаем прибор. Событие «освобождение прибора» также происходит в некоторой точке времени.

Рассмотрим подробнее 2-й и 3-й пункты. Чтобы занять прибор, транзакт должен войти в некоторый блок, который должен обладать следующими свойствами:

- 1) если прибор уже используют, транзакт не может войти в блок, т.е. ему не разрешается занять прибор в данный момент времени, и он должен ждать в очереди - т.е. таким образом транзакт на время исключается из рассмотрения;
- 2) если прибор не используют, транзакт может войти в блок. Вход транзакта в блок вызывает выполнение подпрограммы обработки этого блока. Результат - изменение статуса прибора из «незанято» в «занято».

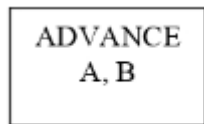
Блоком, который обладает этими свойствами, является блок SEIZE (Занять).

Вход транзакта в блок SEIZE моделирует занятие прибора, вход этого же транзакта в другой блок моделирует освобождение прибора. Этим блоком является блок RELEASE (Освободить).

Оба блока используют операнд A - имя занимаемого или освобождаемого прибора.



Для имитирования работы прибора и, соответственно, задержки транзакта на некоторое время используется блок ADVANCE.



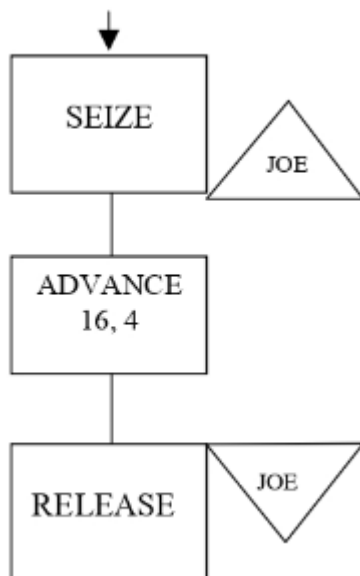
Изображается в виде прямоугольника, располагается между блоками SEIZE и RELEASE (классический случай).

Операнды A и B:

A - задержка на время обслуживания;

B - половина поля допуска.

Классический случай «занять-задержать-освободить»:



Поместив эти три блока между блоками GENERATE и TERMINATE, получим схему продвижения транзакта через прибор.

Написание программы

При написании программы строка экрана условно делится на три части.

Со 2 по 6 позицию записывается имя блока, с 8-й по 18 позицию операции.

Операнды записываются в поле, начинающемся с позиции 19. Операнды

следуют один за другим, их отделяют друг от друга запятыми. Между операндами не должно быть пробелов. Первый пробел, содержащийся в поле операндов, указывает на то, что операнды закончены и интерпретатор прекращает чтение строки. Это свойство может быть использовано для записи поясняющих комментариев.

Первая позиция строки не является частью поля имени. Если в этой позиции стоит символ `*`, то интерпретатор игнорирует всю строку. Это дает возможность документировать модель с помощью более длинных комментариев, а также позволяет оставлять пропуск между отдельными сегментами модели.

Текст программы всегда начинается со слова `SIMULATE` (МОДЕЛИРОВАТЬ), помещенного в поле операций.

Затем следует текст программы, т.е. описание блоков сегментов модели; сегменты отделяются друг от друга строками комментариев.

За описанием блоков следуют управляющие команды. Из них две присутствуют обязательно - это команды `START` и `END`.

Моделирование начинается после того, как интерпретатором в модели найдена команда `START`. Это объясняет, почему команда располагается в конце программы-модели.

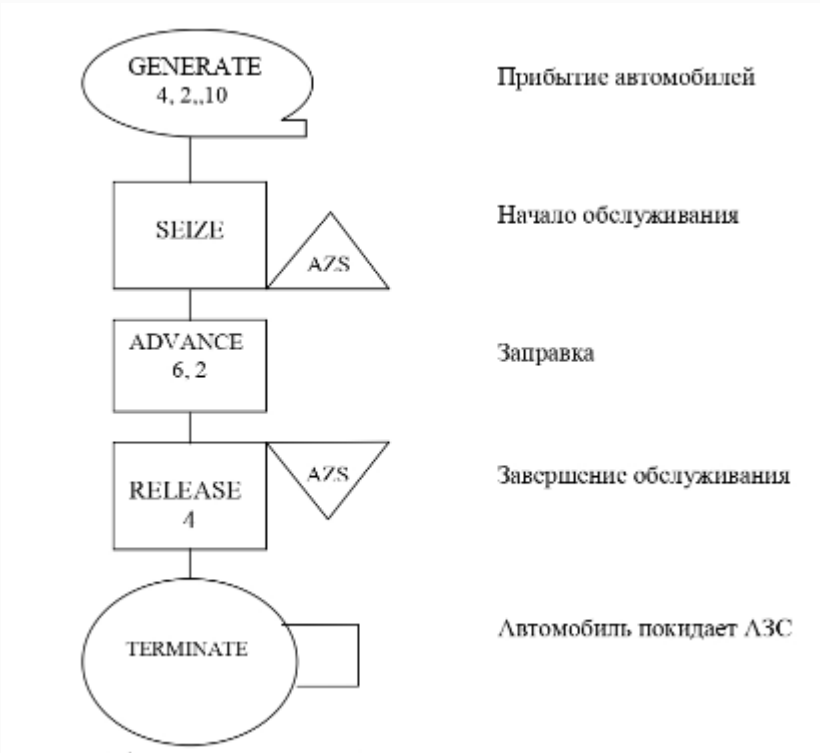
Команда `START` записывается в поле операций и имеет операнд `A`, указывающий количество прогонов модели. Операнд `A` называют еще счетчиком завершений.

После того, как один прогон модели завершен, работа ЭВМ над моделью еще не заканчивается. У разработчика остается много возможностей для продолжения работы. Независимо от того, используются ли эти возможности, существует некоторая точка в модели, перед которой помещены все команды управления прогонами. Здесь должна быть расположена команда `END` (ЗАКОНЧИТЬ). Эта команда побуждает интерпретатор вернуть управление операционную систему. Запись команды осуществляется в поле операции.

Для модели в целом очень важен порядок, в котором располагаются записи блоков отдельных сегментов, но порядок их расположения безразличен.

Пример.

1. Постановка задачи. Обслуживание автомобилей на автозаправочной станции производится с интервалом (6 ± 2) мин. Клиенты приходят с интервалом (4 ± 2) мин. Смоделируем обслуживание десяти машин.



3. Таблица определений.

Сегмент 1

Единица моделирования - 1 мин.

Элементы GPSS	Назначение
Транзакты: 1-й сегмент Приборы: АЗС	Автомобили Заправочная станция

4. Текст программы.

```

SIMULATE
*
*   1-й сегмент
*
GENERATE      4, 2
SEIZE         AZS
ADVANCE       6, 2
RELEASE       AZS
TERMINATE     1
*
*   Команды управления
*
START         300
END
```

Варианты заданий

1. Заказы на горячие блюда в кафе поступают с интервалом (t_1) мин., время на приготовление блюда занимает от 20 до 10 мин. Смоделируйте обслуживание 20 клиентов.

2. Прививка животных ветеринаром занимает (t_2) мин., животные поступают с интервалом (t_3) мин. Смоделируйте очередь из 70 животных.

3. Обслуживание автомобилей на автозаправочной станции производится в среднем за (6 ± 2) мин., клиенты приходят с интервалом (4 ± 2) мин. Смоделируйте очередь для (k_4) автомобилей.

4. Прием у врача длится от 10 до 15 мин., пациенты приходят с интервалом (t_5) мин. За время приема врач обслуживает 20 пациентов.

5. Поиск книги в библиотеке занимает от 5 до 7 мин., читатель обращается за книгами каждые (t_6) мин. Смоделируйте обслуживание 50 человек.

6. Заявка на резервирование билетов поступают с интервалом (5 ± 2) мин., времени на одну операцию затрачивается (t_7) мин. Смоделировать очередь из 30 человек.

7.Экзаменатор принимает экзамен у одного человека за (t_8) мин., группа состоит из 25 студентов. Смоделируйте очередь, если студенты готовятся к ответу (k_9) мин.

8.Токарь обрабатывает деталь на станке за (12 ± 3) мин., детали приходят с интервалом (7 ± 2) мин. Смоделируйте очередь для 100 деталей.

№	t_1	t_2	t_3	k_4	t_5	t_6	t_7	t_8	k_9
1.	7 ± 3	5 ± 2	2 ± 1	300	3 ± 1	7 ± 3	12 ± 10	15 ± 5	25
2.	8 ± 3	5 ± 2	2 ± 1	300	3 ± 1	8 ± 2	12 ± 10	15 ± 5	25
3.	8 ± 3	6 ± 2	2 ± 1	300	3 ± 1	8 ± 2	12 ± 9	15 ± 5	25
4.	9 ± 3	6 ± 2	3 ± 1	300	3 ± 1	8 ± 2	12 ± 9	14 ± 4	25
5.	9 ± 3	6 ± 2	3 ± 1	250	3 ± 1	8 ± 2	12 ± 9	14 ± 4	30
6.	9 ± 3	6 ± 2	3 ± 1	250	4 ± 1	8 ± 2	12 ± 9	14 ± 4	30
7.	9 ± 3	6 ± 2	3 ± 1	250	4 ± 1	6 ± 2	12 ± 9	14 ± 4	30
8.	9 ± 3	6 ± 2	3 ± 1	250	4 ± 1	6 ± 2	13 ± 9	14 ± 4	30
9.	9 ± 3	6 ± 2	3 ± 1	250	4 ± 1	6 ± 2	13 ± 9	16 ± 4	30
10.	9 ± 3	6 ± 2	3 ± 1	250	4 ± 1	6 ± 2	13 ± 9	16 ± 4	20
11.	8 ± 4	6 ± 2	3 ± 1	250	4 ± 1	6 ± 2	13 ± 9	16 ± 4	20
12.	8 ± 4	6 ± 3	3 ± 1	250	4 ± 1	6 ± 2	13 ± 9	16 ± 4	20
13.	8 ± 4	6 ± 3	3 ± 2	250	4 ± 1	6 ± 2	13 ± 9	16 ± 4	20
14.	8 ± 4	6 ± 3	3 ± 2	270	4 ± 1	6 ± 2	13 ± 9	16 ± 4	20
15.	8 ± 4	6 ± 3	3 ± 2	270	4 ± 2	6 ± 3	13 ± 9	16 ± 4	20