

# Язык имитационного моделирования GPSS

Моделирование информационных процессов

мехмат, III курс, группа 11

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

# Основные концепции системы GPSS

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

**GPSS:** (*General Purpose Systems Simulator* — общецелевая система моделирования) — язык программирования, используемый для имитационного моделирования систем (в основном, массового обслуживания). Разработан в 1961 г.

Назначение системы: имитационное моделирование сложных дискретных и непрерывных систем.

Области применения:

- системы массового обслуживания (*Q-схемы*);
- конечные и вероятностные автоматы (*F-* и *P-схемы*);
- сети Петри (*N-*, *N<sub>S</sub>-схемы*, и т. д.);
- И т. д.

Предоставляемое средство: язык имитационного моделирования GPSS.

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Объекты системы GPSS создаются в среде GPSS World и могут быть сохранены в файлах. Основными объектами являются:

**Модель:** содержит описание в виде последовательности *операторов*, а также множество настроек, закладок в тексте программы, сообщений об ошибках и т. д.

**Эксперимент:** создаётся в результате *трансляции* модели, далее выполнение может осуществляться по этапам под действием *команд* (из описания модели, либо вводятся пользователем).

**Отчёт:** создаётся при (успешном) завершении выполнения эксперимента. Содержит статистику по модели в целом и отдельным блокам.

**Текст:** содержит часть описания модели (подключаются в тексте программы, использоваться при вводе/выводе и т. д.).

- Концепции
- Назначение
- Объекты
- Операторы
- Формат
- Транзакция
- Действия
- Пример 1
- Отчёт 1
- Пояснения 1
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Заключение

Все операторы языка GPSS делятся на *блоки* и *команды* (см. рис. 5).

**Блок:** элемент модели, выполняющий определённые функции.

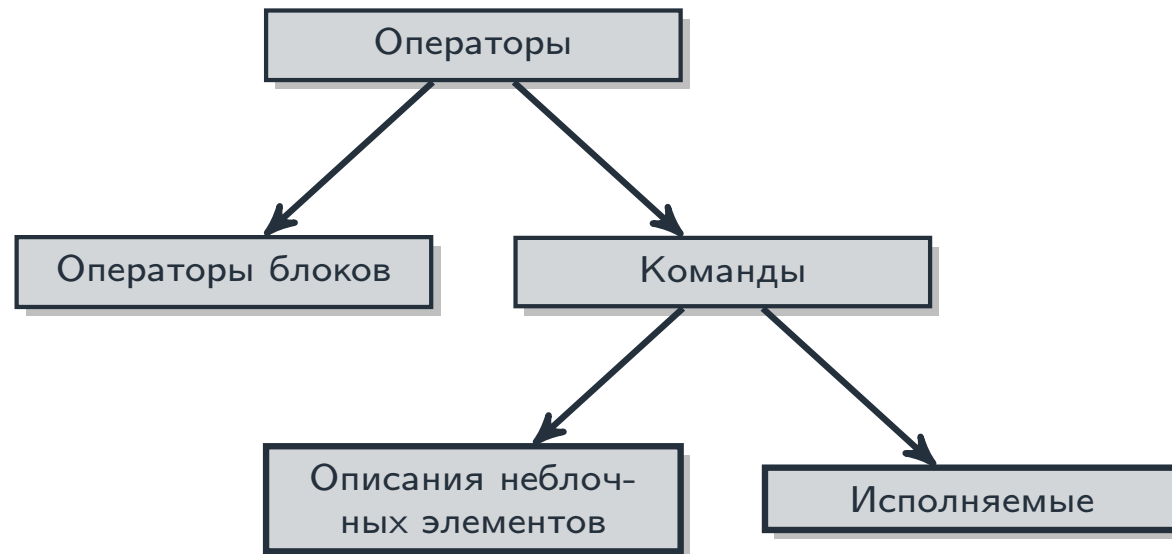


Рис. 1: Классификация операторов GPSS

- Концепции
- Назначение
- Объекты
- Операторы
- Формат
- Транзакция
- Действия
- Пример 1
- Отчёт 1
- Пояснения 1
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Заключение

## Формат оператора:

```
[<№ строки>] [<метка>] <имя> [<операнды>] [<комментарий>]
```

## Пример простой программы:

```
***** Имитация работы парикмахерской *****  
GENERATE 300,100 ; Создание потока посетителей  
SEIZE Barber ; Захват или ожидание парикмахера  
ADVANCE 400,200 ; Обслуживание клиента  
RELEASE Barber ; Освобождение парикмахера  
TERMINATE 1 ; Уход клиента  
* Исполняемые команды  
START 100 ; Имитация 100 клиентов
```

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

**Транзакция:** (сообщение) динамический объект, создаётся в процессе эксперимента в определённых точках модели, продвигается через блоки и затем уничтожается.

Атрибуты, связанные с транзакцией:

- номер (1, 2, ...);
- время появления в системе;
- номер текущего блока;
- состояние (активное, пассивное, завершённое, и т. д.);
- и т. д.
- набор пользовательских параметров («номер или имя» — «значение»).

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

- Одновременно в разных блоках модели могут находиться различные транзакции.
- Транзакции перемещаются по блокам модели в направлении увеличения номеров строк программы, описывающих блоки, если только их направление не изменяется под действием специальных блоков.
- Все действия над транзакциями выполняются мгновенно с точки зрения модельного времени — за исключением явных задержек в специальных блоках, а также ожидания определённых событий.
- В блоках над транзакциями могут выполняться следующие действия:
  - создание транзакций;
  - уничтожение транзакций;
  - изменение параметров транзакций;
  - задержка транзакций на определённый период времени;
  - изменение маршрута транзакции в модели.



# Создание и запуск простой программы в среде GPSS World

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

1. Запустить среду GPSS World.
2. Выбрать команду меню **File** → **New**, в появившемся окне выбрать **Model**:

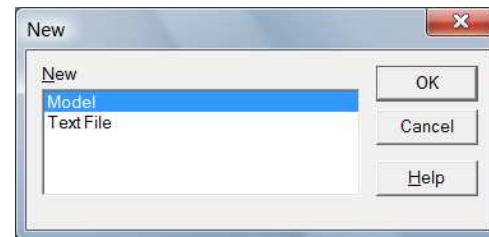


Рис. 2: Окно выбора типа документа

3. В появившемся редакторе набрать следующую программу:

```
; my0.gps  
GENERATE 60           ; Создание транзакций  
TERMINATE 1          ; Уничтожение транзакций  
* Команды  
START 100           ; Моделирование 100 транзакций
```

4. Выбрать команду меню **Command** → **Create Simulation**.

- Концепции
- Назначение
- Объекты
- Операторы
- Формат
- Транзакция
- Действия
- Пример 1
- Отчёт 1
- Пояснения 1
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Заключение

```

GPSS World Simulation Report - Untitled Model 7.1.1

Friday, February 19, 2010 03:48:20

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000              6000.000    2        0           0

LABEL              LOC  BLOCK TYPE      ENTRY COUNT  CURRENT COUNT  RETRY
1      GENERATE          100           0           0
2      TERMINATE         100           0           0

FEC XN  PRI      BDT      ASSEM  CURRENT  NEXT  PARAMETER  VALUE
101     0      6060.000  101    0        1
    
```

## Концепции

Назначение

Объекты

Операторы

Формат

Транзакция

Действия

Пример 1

Отчёт 1

Пояснения 1

## Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

- Поля START TIME и END TIME указывают моменты модельного времени начала и конца моделирования.
- Поле BLOCKS указывает количество блоков, составляющих модель (в данном примере — блоки GENERATE и TERMINATE).
- Поля FACILITIES и STORAGES указывают количество устройств обслуживания заявок и накопителей, присутствующих в модели.
- В следующей таблице представлена информация о блоках модели. Колонка LABEL содержит метки блоков.
- Колонка LOC содержит номера блоков, назначаемые системой.
- Колонка BLOCK TYPE содержит имена блоков.
- Колонка ENTRY COUNT для каждого блока содержит количество транзакций, вошедших в данный блок за всё время моделирования.
- Колонка CURRENT COUNT для каждого блока содержит количество транзакций, задержанных в блоке на момент окончания моделирования.

Концепции

Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

## Основные блоки и операторы GPSS

# Блоки создания и удаления транзакций

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Данный блок определяет точку входа транзакций в модель вместе с вероятностными характеристиками входящего потока. Попытка входа в данный блок другой транзакции («сверху») приводит к ошибке.

```
GENERATE [интервал],[разброс],[задержка],[ограничение],[приоритет]
```

интервал	—	среднее время между созданием транзакций;
разброс	—	разброс времени между созданием транзакций (число или функция);
задержка	—	начальная задержка создания первой транзакции;
ограничение	—	максимальное количество создаваемых транзакций;
приоритет	—	приоритет создаваемых транзакций.

Данный блок уничтожает транзакции, входящие в него. Попытка захвата транзакцией блока всегда успешна.

```
TERMINATE [вычитаемое]
```

вычитаемое	—	значение, вычитаемое из счётчика завершения при удалении очередной транзакции (0 по умолчанию).
------------	---	---

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

Данная команда предназначена для запуска имитации модели. Её можно ввести либо из интегрированной среды GPSS World (команда меню **Command** → **START**, которая становится доступной после трансляции модели при помощи **Command** → **Create Simulation**), либо непосредственно указать в тексте программы. Тогда запуск модели выполнится сразу после её (успешной) трансляции.

```
START <начальное значение>
```

начальное значение — начальное значение, присваиваемое счётчику завершения.

# Пример моделирования одноканального устройства

Рассмотрим пример простой системы со входящим потоком заявок, накопителем неограниченной ёмкости, одноканальным устройством обслуживания и точкой выхода заявок (см. рис. 3).

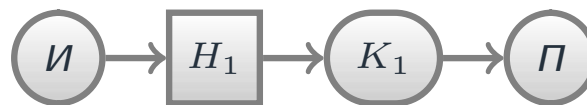


Рис. 3: Система массового обслуживания с одноканальным устройством

Данную модель описывает следующая программа:

```
* Блоки модели
    GENERATE 7,2           ; Создание транзакций
    SEIZE Facility1       ; Захват устройства
    ADVANCE 5,2           ; Задержка транзакций
    RELEASE Facility1     ; Освобождение устройства
    TERMINATE 1           ; Уничтожение транзакций
* Команды
    START 100             ; Моделирование 100 транзакций
```

В ней при помощи специальных блоков (см. далее) определяется одноканальное устройство обслуживания с именем Facility1. Трансляция программы приводит к её запуску и созданию отчёта (см. далее).

- Концепции
- Основные блоки
  - Создание
  - Запуск
  - Пример 2
  - Отчёт 2
  - Пояснения 2
    - Захват
    - Задержка
    - Очередь
  - Пример 3
  - Пояснения 3.1
  - Отчёт 3
  - Пояснения 3.2
  - Устройство 2
  - Пример 4.1
  - Пример 4.2
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Заключение

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	714.047	5	1	0

NAME	VALUE
FACILITY1	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	SEIZE	100	0	0
	3	ADVANCE	100	0	0
	4	RELEASE	100	0	0
	5	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	100	0.699	4.994	1	0	0	0	0	0

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
101	0	718.106	101	0	1		



## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

Недоступность

Циклы

Ситуации

Заключение

- В данном отчёте по сравнению с предыдущим появилась таблица с информацией обо всех устройствах модели. В колонке FACILITY перечислены имена всех устройств модели.
- В колонке ENTRIES указывается количество транзакций, прошедших через данное устройство за время моделирования.
- В колонке UTIL. указывается коэффициент использования устройства: доля модельного времени, в течение которого оно было занято.
- В колонке AVE. TIME указывается среднее время обработки одной транзакции в заданном устройстве.
- В колонке AVAIL. указывается состояние устройства в момент окончания моделирования (1 — свободно; 0 — занято).
- В колонке OWNER указывается номер транзакции, занимавшей устройство в момент окончания моделирования (0 — устройство было свободно).
- В колонке PEND указывается количество транзакций, ожидающих освобождения устройства.

# Блок захвата одноканального устройства

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

Данный блок устанавливает заданное одноканальное устройство в состояние занятости входящей в него транзакцией.

```
SEIZE <имя устройства>
```

имя устройства — имя или номер устройства.

Действие блока:

- Если данное устройство *свободно*, то входящая транзакция *занимает* его и пытается войти в следующий блок.
- Если устройство *занято*, транзакция помещается последней в *очередь* ожидающих это устройство (с тем же приоритетом).

# Блоки задержки и освобождения транзакции

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

Прохождение транзакций через большинство блоков осуществляется мгновенно с точки зрения моделирования. Внесение задержек в прохождение осуществляется при помощи специальных блоков.

```
ADVANCE [интервал], [разброс]
```

*интервал*, *разброс* — имеют тот же смысл, что и аналогичные параметры блока **GENERATE**.

```
RELEASE имя устройства
```

Действие блока:

- Транзакция, владеющая устройством, покидает его и переходит к следующему блоку.
- Транзакция, вытесненная (**PREEMPT**) из владения устройством, удаляется из его списка прерывания.

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Данные блоки предназначены для сбора статистики об очереди (по соображениям скорости моделирования не для всех очередей, образующихся в модели, ведётся статистика). На поведение модели данные блоки никак не влияют.

```
QUEUE <имя очереди> [, <количество>]
```

**количество** — количество единиц, на которые увеличивается длина очереди при вхождении заявки в блок.

```
DEPART <имя очереди> [, <количество>]
```

**количество** — количество единиц, на которые уменьшается длина очереди при выходе заявки из блока.

# Пример сбора статистики об очереди

## Концепции

### Основные блоки

- Создание
- Запуск
- Пример 2
- Отчёт 2
- Пояснения 2
- Захват
- Задержка
- Очередь
- Пример 3
- Пояснения 3.1
- Отчёт 3
- Пояснения 3.2
- Устройство 2
- Пример 4.1
- Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

#### \* Блоки модели

<b>GENERATE</b>	5,2	; Создание транзакций
<b>QUEUE</b>	Queue1	; Вход транзакций в очередь
<b>SEIZE</b>	Facility1	; Захват устройства
<b>DEPART</b>	Queue1	; Выход транзакций из очереди
<b>ADVANCE</b>	7,2	; Задержка транзакций
<b>RELEASE</b>	Facility1	; Освобождение устройства
<b>TERMINATE</b>	1	; Уничтожение транзакций

#### \* Команды

<b>START</b>	100	; Моделирование 100 транзакций
--------------	-----	--------------------------------

Этот пример демонстрирует использование блоков **QUEUE** и **DEPART** для сбора статистики об очереди одноканального устройства.

# Пояснения к примеру сбора статистики об очереди

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

Недоступность

Циклы

Ситуации

Заключение

- Прежде чем попытаться занять устройство Facility1 (блок **SEIZE**), транзакция проходит через блок **QUEUE**, увеличивая счётчик для статистики Queue1.
- Если устройство Facility1 в текущий модельный момент занято, транзакция остаётся в его очереди, не покидая блока **SEIZE** (и не меняя статистики Queue1).
- Как только устройство Facility1 освобождается, из очереди в блоке **SEIZE** выбирается очередная транзакция и перемещается в следующий за ним блок. Данным блоком является блок **DEPART**, уменьшающий при этом счётчик для статистики Queue1.
- После прохождения блока **DEPART** транзакция переходит в блок **ADVANCE**, имитирующий задержку при её обработке на устройстве Facility1.

## Концепции

### Основные блоки

- Создание
- Запуск
- Пример 2
- Отчёт 2
- Пояснения 2
- Захват
- Задержка
- Очередь
- Пример 3
- Пояснения 3.1
- Отчёт 3
- Пояснения 3.2
- Устройство 2
- Пример 4.1
- Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	142	0	0
	2	QUEUE	142	41	0
	3	SEIZE	101	1	0
	4	DEPART	100	0	0
	5	ADVANCE	100	0	0
	6	RELEASE	100	0	0
	7	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	101	0.991	7.015	1	101	0	0	0	41

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
QUEUE1	42	42	142	1	21.968	110.599	111.383 0

- В сгенерированном отчёте данного примера информация в колонках ENTRY COUNT и CURRENT COUNT говорит о том, что на момент окончания моделирования (прохождения 100 заявок через систему), в некоторых блоках оставались заявки, не успевшие выйти из системы.
- В данном отчёте по сравнению с предыдущим появилась таблица с информацией обо всех очередях модели.

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

- В колонке QUEUE перечислены имена очередей.
- В колонке MAX указывается максимальная длина очереди за время моделирования.
- В колонке CONT. указывается длина очереди на момент окончания моделирования.
- В колонке ENTRY указывается общее количество входов транзакций в очередь за всё время моделирования.
- В колонке ENTRY(0) указывается общее количество входов транзакций в очередь, имеющих нулевое время ожидания в ней, за всё время моделирования.
- В колонке AVE.CONT. указывается средняя длина очереди за время моделирования.
- В колонке AVE.TIME указывается среднее время ожидания в очереди за время моделирования.
- Аналогично, в колонке AVE.(-0) указывается среднее время ожидания в очереди без учёта заявок, имевших в ней нулевое время ожидания.



# Оператор и блоки определения многоканального устройства

## Концепции

### Основные блоки

Создание

Запуск

Пример 2

Отчёт 2

Пояснения 2

Захват

Задержка

Очередь

Пример 3

Пояснения 3.1

Отчёт 3

Пояснения 3.2

Устройство 2

Пример 4.1

Пример 4.2

### Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Данный оператор позволяет определить ёмкость (количество обслуживающих линий) многоканального устройства:

```
<имя> STORAGE <ёмкость>
```

имя — имя устройства, для которого определяется ёмкость;  
ёмкость — значение ёмкости.

Данный блок определяет точку входа в многоканальное устройство:

```
ENTER <имя>, [<количество>]
```

имя — имя или номер многоканального устройства;  
количество — количество каналов устройства, занимаемых транзакцией (1 по умолчанию).

Данный блок определяет освобождение каналов многоканального устройства:

```
LEAVE <имя>, [<количество>]
```

# Модель с многоканальным устройством обслуживания

- Концепции
- Основные блоки
  - Создание
  - Запуск
  - Пример 2
  - Отчёт 2
  - Пояснения 2
  - Захват
  - Задержка
  - Очередь
  - Пример 3
  - Пояснения 3.1
  - Отчёт 3
  - Пояснения 3.2
  - Устройство 2
  - Пример 4.1
  - Пример 4.2
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Заключение

Рассмотрим теперь пример системы с трёхканальным устройством обслуживания (см. рис. 4).

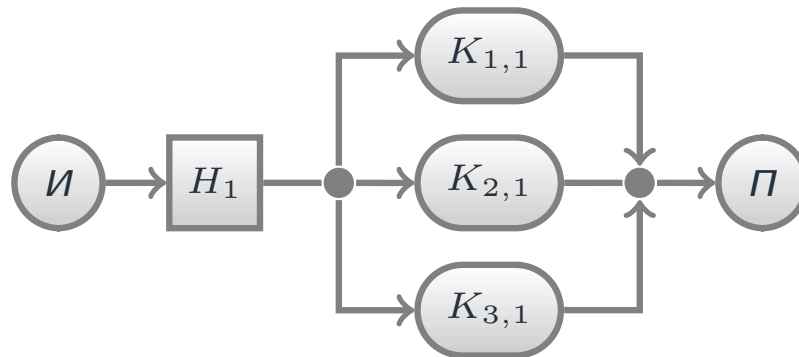


Рис. 4: Система массового обслуживания с многоканальным устройством

Эту модель можно описать при помощи программы (см. далее).

# Моделирование многоканального устройства

## Концепции

### Основные блоки

Создание  
Запуск  
Пример 2  
Отчёт 2  
Пояснения 2  
Захват  
Задержка  
Очередь  
Пример 3  
Пояснения 3.1  
Отчёт 3  
Пояснения 3.2  
Устройство 2  
Пример 4.1  
Пример 4.2

### Перенаправление

### Недоступность

### Циклы

### Ситуации

### Заключение

#### \* *Описания*

Multi1 STORAGE 3 ; Вместимость устройства

#### \* *Блоки модели*

GENERATE 3,2 ; Создание транзакций  
QUEUE Queue1 ; Вход транзакций в очередь  
ENTER Multi1 ; Захват транзакций  
DEPART Queue1 ; Выход транзакций из очереди  
ADVANCE 12,2 ; Задержка транзакций  
LEAVE Multi1 ; Освобождение транзакций  
TERMINATE 1 ; Уничтожение транзакций

#### \* *Команды*

START 100 ; Моделирование 100 транзакций

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

## Блоки перенаправления транзакций

# Блок перехода на свободное устройство, безусловный переход

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

Данный вариант блока осуществляет переход в первый из двух доступных блоков:

```
TRANSFER BOTH, [⟨метка1⟩], ⟨метка2⟩
```

- метка<sub>1</sub> — метка блока, в который осуществляется переход, если он свободен (по умолчанию, следующий блок);
- метка<sub>2</sub> — метка блока, в который осуществляется переход, если он свободен и первый блок занят.

Данный блок осуществляет безусловное перенаправление транзакции на блок по указанной метке:

```
TRANSFER ,⟨метка⟩
```

Пример безусловного перехода:

```
TRANSFER ,Finish
```

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

Данный блок осуществляет перенаправление транзакции на два заданных блока с заданной вероятностью:

```
TRANSFER <вероятность>, [<метка1>], <метка2>
```

- |                    |   |   |
|--------------------|---|---|
| вероятность        | — | вещественное выражение — вероятность перехода транзакции по метке «метка <sub>2</sub> »;                  |
| метка <sub>1</sub> | — | метка блока, в который осуществляется перенаправление каждой транзакции с вероятностью 1 — «вероятность»; |
| метка <sub>2</sub> | — | метка блока, в который осуществляется перенаправление каждой транзакции с вероятностью «вероятность».     |

Пример перехода по вероятности:

```
TRANSFER .75,Place_0_25,Place_0_75
```

# Моделирование очереди ограниченной ёмкости

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

Рассмотрим применение блоков `ENTER`, `LEAVE` и `TRANSFER` для моделирования очередей ограниченной ёмкости. Пусть дан пример системы с одноканальным устройством обслуживания, аналогичной системе на рис. 3. Однако теперь пусть накопитель  $H_1$  будет иметь ограниченную ёмкость в 3 заявки. Данную модель можно описать при помощи следующей программы:

```
* Описания
Seats STORAGE 3 ; Вместимость очереди
* Блоки модели
* Входящие заявки
GENERATE 4,2 ; Создание транзакций
TRANSFER BOTH, ,Lost ; Уход, если очередь занята
* Канал обслуживания с очередью
ENTER Seats ; Вход в накопитель
SEIZE Barber ; Захват канала
LEAVE Seats ; Выход из накопителя
ADVANCE 13,3 ; Задержка обслуживания
RELEASE Barber ; Освобождение канала
* Заявки, прошедшие/не прошедшие канал
TERMINATE 1 ; Уничтожение транзакций
Lost TERMINATE 1 ; Уничтожение транзакций
* Команды
START 100 ; Моделирование 100 транзакций
```

# Пояснения к примеру моделирования очереди ограниченной ёмкости

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

- Каждая транзакция, выходящая из блока **GENERATE**, будет попадать в блок **TRANSFER**, из которого сначала будет пытаться переходить в следующий блок (пустой второй операнд), если устройство, определяемое им, будет свободно.
- Следующим блоком является блок **ENTER**, определяющий многоканальное устройство `seats`, которое в данном примере предназначено для моделирования очереди, а не устройства обслуживания. Если в текущий момент модельного времени все 3 канала (ёмкость определяется оператором **STORAGE**) устройства заняты, транзакция не попадёт в блок **ENTER**. Вместо этого она попытается из блока **TRANSFER** перейти по метке `lost` в блок **TERMINATE**, что будет всегда успешным.
- Иначе (в устройстве `seats` будут свободные каналы) транзакция попадёт в блок **ENTER**, уменьшив счётчик свободных каналов для устройства `seats`.
- Далее транзакция перейдёт в следующий блок **SEIZE** и попытается перейти дальше, что будет успешным, если одноканальное устройство `barber` в данный момент будет свободным.



# Пояснения к примеру моделирования очереди ограниченной ёмкости (окончание)

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

- Если всё же устройство `barber` занято, транзакция останется в очереди, связанной с блоком `SEIZE`. Длина этой очереди никогда не превысит 3 транзакции, так как находящиеся в ней транзакции уменьшили счётчик свободных каналов устройства `seats`.
- Иначе (устройство `barber` свободно) транзакция из блока `SEIZE` перейдёт дальше, при этом устройство `barber` будет отмечено как занято. Следующим блоком будет блок `LEAVE`, увеличивающий счётчик свободных каналов устройства `seats` (таким образом, транзакция освободит место в очереди).
- Далее транзакция попадёт в блок `ADVANCE`, моделирующий временную задержку при обслуживании в устройстве `barber`.
- Далее транзакция перейдёт в блоки `RELEASE` и `TERMINATE`. Причём в последний будут попадать только транзакции, прошедшие через устройство `barber`, поскольку потерянные заявки будут направлены в другой блок `TERMINATE` (с меткой `lost`). Это позволяет собирать отдельно статистику для обслуженных и потерянных заявок.

- Концепции
- Основные блоки
- Перенаправление
- Переход 1
- Переход 2
- Пример 5
- Пояснения 5.1
- Пояснения 5.2
- Отчёт 5
- Пояснения 5.3
- Выводы 5
- Переход 3
- Пример 6
- Отчёт 6
- Переход 4
- Пример 7
- Отчёт 7
- Недоступность
- Циклы
- Ситуации
- Заключение

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES					
	0.000	417.107	9	1	1					
	...									
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY				
	1	GENERATE	104	0	0	0				
	2	TRANSFER	104	0	0	0				
	3	ENTER	34	3	0	0				
	4	SEIZE	31	0	0	0				
	5	LEAVE	31	0	0	0				
	6	ADVANCE	31	1	0	0				
	7	RELEASE	30	0	0	0				
	8	TERMINATE	30	0	0	0				
LOST	9	TERMINATE	70	0	0	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
BARBER	31	0.987	13.282	1	94	0	0	0	3	
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
SEATS	3	0	0	3	34	1	2.774	0.925	0	0

## Концепции

### Основные блоки

#### Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

### Недоступность

### Циклы

### Ситуации

### Заключение

- По информации из колонки `ENTRY COUNT` за моделируемое время в системе успешно прошли обработку 30 заявок (блок `TERMINATE` под номером 8) и покинули её из-за переполнения очереди 70 заявок (следующий блок `TERMINATE` под номером 9). Всего из системы вышло  $30 + 70 = 100$  заявок (условие останова моделирования, определяемое в команде `START`).
- По информации из колонки `CURRENT COUNT` на момент окончания моделирования в системе оставалось 3 заявки в очереди (блок `ENTER` под номером 3) и одна заявка в устройстве обслуживания (блок `ADVANCE` под номером 6).
- По информации из колонки `UTIL.` таблицы устройств коэффициент загрузки устройства составлял 0,987, что близко к 1 (постоянная занятость).
- По информации из колонок `AVE.C.` и `UTIL.` таблицы хранилищ средняя длина очереди составляла 2,774 единицы, а коэффициент загрузки — 0,925, что также близко к постоянной занятости.

## Концепции

### Основные блоки

#### Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

**Выводы 5**

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

#### Недоступность

#### Циклы

#### Ситуации

#### Заключение

*Замечание:* доля успешно обслуженных заявок:  $30 \div 100 = 0,3$  очень близка к отношению средних интервалов поступления в систему и обслуживания заявок:  $4 \div 13 \approx 0,308$ . Эксперименты показывают, что увеличение ёмкости очереди до 30 приводит к увеличению количества успешно обслуженных заявок всего лишь до 37. △

# Блок перехода в первый из нескольких блоков

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

Данный блок аналогичен блоку **TRANSFER BOTH** за исключением того, что позволяет указать диапазон блоков (вместо двух), отстоящих друг от друга в описании модели на фиксированное количество блоков.

```
TRANSFER ALL, <метка1>[, <меткаn>][, <количество блоков>]
```

- |                    |   |   |
|--------------------|---|---|
| метка <sub>1</sub> | — | метка первого блока;  |
| метка <sub>n</sub> | — | метка последнего блока (последний ~ первому, если не указана);  |
| количество блоков  | — | расстояние между «соседними» блоками в блоках (1 по умолчанию). |

# Пример перенаправления транзакций в первое из нескольких устройств

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

```
GENERATE 15
TRANSFER ALL,First,Last,4
* I устройство
First SEIZE Facility1
ADVANCE 12,4
RELEASE Facility1
TRANSFER ,Finish
* II устройство
SEIZE Facility2
ADVANCE 40
RELEASE Facility2
TRANSFER ,Finish
* III устройство
Last SEIZE Facility3
ADVANCE 20
RELEASE Facility3
* Выход из системы
Finish TERMINATE 1
* Запуск модели
START 100
```

- Концепции
- Основные блоки
- Перенаправление
- Переход 1
- Переход 2
- Пример 5
- Пояснения 5.1
- Пояснения 5.2
- Отчёт 5
- Пояснения 5.3
- Выводы 5
- Переход 3
- Пример 6
- Отчёт 6
- Переход 4
- Пример 7
- Отчёт 7
- Недоступность
- Циклы
- Ситуации
- Заключение

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	101	0	0	
	2	TRANSFER	101	0	0	
FIRST	3	SEIZE	86	0	0	
	4	ADVANCE	86	1	0	
	5	RELEASE	85	0	0	
	6	TRANSFER	85	0	0	
	7	SEIZE	14	0	0	
	8	ADVANCE	14	0	0	
	9	RELEASE	14	0	0	
	10	TRANSFER	14	0	0	
LAST	11	SEIZE	1	0	0	
	12	ADVANCE	1	0	0	
	13	RELEASE	1	0	0	
FINISH	14	TERMINATE	100	0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	86	0.678	12.021	1	101	0	0	0	0
FACILITY2	14	0.367	40.000	1	0	0	0	0	0
FACILITY3	1	0.013	20.000	1	0	0	0	0	0

# Блок случайного перехода в один из нескольких блоков

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

Данный блок перенаправляет каждую входящую в него транзакцию в один из блоков, расположенных между заданными метками, случайным образом (с одинаковой вероятностью).

```
TRANSFER PICK, <метка1>, <меткаn>
```

метка<sub>1</sub> — метка первого блока;

метка<sub>n</sub> — метка последнего блока.

*Замечание:* для блока **TRANSFER ALL** блоки, указываемые в качестве направления перехода (между метками «метка<sub>1</sub>» и «метка<sub>n</sub>»), могут располагаться на некотором расстоянии друг от друга. Этим данный блок отличается от блока **TRANSFER PICK**, для которого блоки-направления должны находиться непосредственно друг за другом. Если есть необходимость разместить между ними другие блоки, можно воспользоваться блоками безусловного перехода (см. далее). △



# Пример перенаправления транзакций на случайное устройство

Концепции

Основные блоки

Перенаправление

Переход 1

Переход 2

Пример 5

Пояснения 5.1

Пояснения 5.2

Отчёт 5

Пояснения 5.3

Выводы 5

Переход 3

Пример 6

Отчёт 6

Переход 4

Пример 7

Отчёт 7

Недоступность

Циклы

Ситуации

Заключение

```
GENERATE 5,,100
* Перенаправление заявки
TRANSFER PICK,Pick1,PickN
Pick1 TRANSFER ,Place1
      TRANSFER ,Place2
      TRANSFER ,Place3
PickN  TRANSFER ,Place4
* Устройство 1.1
Place1 SEIZE Facility11
      ADVANCE 10
      RELEASE Facility11
      TRANSFER ,Finish
* Устройство 1.2
Place2 SEIZE Facility12
      ADVANCE 10
      RELEASE Facility12
      TRANSFER ,Finish
```

```
* Устройство 1.3
Place3 SEIZE Facility13
      ADVANCE 10
      RELEASE Facility13
      TRANSFER ,Finish
* Устройство 1.4
Place4 SEIZE Facility14
      ADVANCE 10
      RELEASE Facility14
* Выход из системы
Finish TERMINATE 1
* Запуск модели
START 100
```

- Концепции
- Основные блоки
- Перенаправление
- Переход 1
- Переход 2
- Пример 5
- Пояснения 5.1
- Пояснения 5.2
- Отчёт 5
- Пояснения 5.3
- Выводы 5
- Переход 3
- Пример 6
- Отчёт 6
- Переход 4
- Пример 7
- Отчёт 7
- Недоступность
- Циклы
- Ситуации
- Заключение

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY			
	1	GENERATE	100	0	0				
	2	TRANSFER	100	0	0				
PICK1	3	TRANSFER	23	0	0				
	4	TRANSFER	29	0	0				
	5	TRANSFER	28	0	0				
PICKN	6	TRANSFER	20	0	0				
PLACE1	7	SEIZE	23	0	0				
	8	ADVANCE	23	0	0				
	9	RELEASE	23	0	0				
	10	TRANSFER	23	0	0				
.....									
PLACE4	19	SEIZE	20	0	0				
	20	ADVANCE	20	0	0				
	21	RELEASE	20	0	0				
FINISH	22	TERMINATE	100	0	0				
.....									
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY13	28	0.549	10.000	1	0	0	0	0	0
FACILITY12	29	0.569	10.000	1	0	0	0	0	0
FACILITY14	20	0.392	10.000	1	0	0	0	0	0
FACILITY11	23	0.451	10.000	1	0	0	0	0	0

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

## Блоки вытеснения заявок и недоступности устройств

Данный блок определяет, что входящие в него транзакции прерывают текущую обработку транзакции на некотором другом устройстве.

```
PREEMPT <устройство>, [режим приоритета], [назначение], [№ параметра], [режим]
```

- |                  |   |   |
|------------------|---|---|
| устройство       | — | на котором происходит прерывание;   |
| режим приоритета | — | <b>PR</b> — режим приоритета (обрабатываемая транзакция может быть прервана только транзакцией с бóльшим приоритетом), либо «пусто» — режим прерывания;   |
| назначение       | — | имя блока, куда направляются транзакции прерванного потока;   |
| № параметра      | — | номер параметра, связанного с прерванной транзакцией, в который будет записано оставшееся время, которое она не успела провести в блоке <b>ADVANCE</b> ;  |
| режим            | — | <b>RE</b> — режим удаления прерванных транзакций (прерванная транзакция не будет претендовать на использование устройства, на котором произошло прерывание, вместо этого будет направлена в блок, заданный полем «назначение»), либо «пусто». |

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

# Блок завершения вытеснения транзакций

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

Данный блок является парным по отношению к блоку **PREEMPT** и определяет, что проходящая через него заявка завершает прерывание обработки на другом устройстве.

```
RETURN <устройство>
```

устройство — на котором завершается прерывание.

# Пример вытеснения заявками II потока заявок из I потока

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

\* Блоки модели

\* Входящие заявки I потока

<b>GENERATE</b>	5,2	; Транзакции I потока
<b>SEIZE</b>	Facility1	; Захват канала
<b>ADVANCE</b>	7,2	; Задержка обслуживания
<b>RELEASE</b>	Facility1	; Освобождение канала
<b>TERMINATE</b>	1	; Уничтожение транзакций

\* Входящие заявки II потока

<b>GENERATE</b>	25,2	; Транзакции II потока
<b>PREEMPT</b>	Facility1,,Lost,,RE	; Прерывание I канала
<b>ADVANCE</b>	11,2	; Задержка обслуживания
<b>RETURN</b>	Facility1	; Освобождение I канала
Lost <b>TERMINATE</b>	1	; Уничтожение транзакций

\* Команды

<b>START</b>	100	; Моделирование 100 транзакций
--------------	-----	--------------------------------

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Вытеснение
- Завершение
- Пример 8
- Отчёт 8
- Устройство
- Транзакции
- Пример 9
- Отчёт 9
- Очередь
- Циклы
- Ситуации
- Заключение

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	143	69	0	
	2	SEIZE	74	1	0	
	3	ADVANCE	73	0	0	
	4	RELEASE	46	0	0	
	5	TERMINATE	46	0	0	
	6	GENERATE	27	0	0	
	7	PREEMPT	27	0	0	
	8	ADVANCE	27	0	0	
	9	RETURN	27	0	0	
LOST	10	TERMINATE	54	0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	101	0.990	6.972	1	89	0	0	0	69

# Блок перевода устройства в недоступное состояние

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

Данный блок моделирует ситуацию, когда некоторое одноканальное устройство (facility) в течение определённых периодов времени становится недоступным («выход из строя»). При входе транзакции в данный блок заданное устройство становится недоступным.

```
FUNAVAIL <имя>, [<режим1>], [<метка1>], [<№>], [<режим2>], [<метка2>], [<режим3>], [<метка3>]
```

имя	—	имя устройства, становящегося недоступным;
режим <sub>1</sub>	—	режим транзакций I типа (см. далее);
метка <sub>1</sub>	—	метка для перехода транзакций I типа;
№	—	номер параметра, в которое помещается время задержки;
режим <sub>2</sub>	—	режим транзакций II типа (см. далее);
метка <sub>2</sub>	—	метка для перехода транзакций II типа;
...		

Данный блок при прохождении через него транзакции делает доступным заданное устройство, отменяя действие блока **FUNAVAIL**. Если устройство на данный момент уже доступно, блок не оказывает никакого действия.

```
FAVAIL <имя>
```



# Классификация транзакций и режимов их обработки на недоступном устройстве (задающихся в блоке FUNAVAIL)

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

## I. Управляющая (занимающая устройство):

- прерывается, продолжает выполнение после освобождения (режим «пусто»);
- переходит в другой блок (режим **RE**);
- продолжает обслуживаться (режим **CO**).

## II. Прерванные (при помощи **PREEMPT**):

- покидают список прерывания (режим «пусто»);
- больше не претендуют на устройство, удаляются из списка задержки (режим **RE**);
- продолжают обслуживаться в порядке приоритета (режим **CO**).

## III. Задержанные.

- ожидают в очереди (режим «пусто»);
- больше не претендуют на устройство, удаляются из списка задержки, не должны пытаться войти в **RETURN** или **RELEASE** (режим **RE**);
- продолжают обслуживаться в порядке приоритета (режим **CO**).

# Пример перевода устройства в недоступное состояние

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

\* *Входящие заявки I потока*

GENERATE 5,2

SEIZE F\_main

ADVANCE 7

RELEASE F\_main

Finish TERMINATE 1

\* *Входящие заявки II потока, прерывающие*

\* *заявки I потока*

GENERATE 25

PREEMPT F\_main

ADVANCE 11

RETURN F\_main

TERMINATE

\* *Перевод в недоступность устройства F\_main*

GENERATE 55

FUNAVAIL F\_main,RE,F\_I,,RE,F\_II,RE,F\_III

ADVANCE 15

FAVAIL F\_main

TERMINATE

\* *Обслуживание заявок, обслуживавшихся*

\* *на момент прерывания*

F\_I RELEASE F\_main

SEIZE F\_for\_managed

ADVANCE 10

RELEASE F\_for\_managed

TRANSFER ,Finish

\* *Обслуживание заявок, вытесненных*

\* *на момент прерывания*

F\_II SEIZE F\_for\_preempted

ADVANCE 11

RELEASE F\_for\_preempted

TRANSFER ,Finish

\* *Обслуживание заявок, находившихся*

\* *в очереди на момент прерывания*

F\_III SEIZE F\_for\_queued

ADVANCE 12

RELEASE F\_for\_queued

TRANSFER ,Finish

\* *Команды*

START 100

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Вытеснение
- Завершение
- Пример 8
- Отчёт 8
- Устройство
- Транзакции
- Пример 9
- Отчёт 9
- Очередь
- Циклы
- Ситуации
- Заключение

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES			
	0.000	725.000	28	4	0			
...								
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY			
	1	GENERATE	142	47	0			
	2	SEIZE	39	0	0			
	3	ADVANCE	39	0	0			
	4	RELEASE	26	0	0			
FINISH	5	TERMINATE	100	0	0			
	6	GENERATE	28	0	0			
	7	PREEMPT	28	0	0			
	8	ADVANCE	28	0	0			
	9	RETURN	22	0	0			
	10	TERMINATE	22	0	0			
	11	GENERATE	13	0	0			
	12	FUNAVAIL	13	0	0			
	13	ADVANCE	13	1	0			
	14	FAVAIL	12	0	0			
	15	TERMINATE	12	0	0			
F_I	16	RELEASE	13	0	0			
...								
FACILITY	ENTRIES	UTIL.	AVE. TIME AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
F_MAIN	67	0.729	7.890	0	0	0	0	3
F_FOR_PREEMPTED	6	0.091	11.000	1	0	0	0	0
F_FOR_MANAGED	13	0.179	10.000	1	0	0	0	0
F_FOR_QUEUED	56	0.924	11.964	1	106	0	0	44

# Блок перевода очереди в недоступное состояние

Данные блоки, аналогично блокам **FUNAVAIL**/**FAVAIL**, при прохождении через них транзакций делают недоступным/доступным заданное многоканальное устройство (storage).

**SUNAVAIL** *<имя>*

**SAVAIL** *<имя>*

Концепции

Основные блоки

Перенаправление

Недоступность

Вытеснение

Завершение

Пример 8

Отчёт 8

Устройство

Транзакции

Пример 9

Отчёт 9

Очередь

Циклы

Ситуации

Заключение

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

## Блоки организации циклов, расщепления/сборки транзакций

# Команда объявления именованного значения и блок цикла

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данная команда предназначена для объявления именованных значений:

```
<имя> EQU <выражение>
```

имя — метка;

выражение — вычисляется, значение сопоставляется с именем.

Данный блок предназначен для повторного направления транзакции на заданный (предыдущий) блок модели заданное количество раз.

Количество повторений определяется значением параметра транзакции.

```
LOOP <№ параметра> [, <метка>]
```

№ параметра — номер параметра, уменьшаемого на 1;

метка — метка блока, куда направляется транзакция, если значение параметра достигает 0.

# Блоки установки параметров транзакций

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данный блок предназначен для установки/изменения параметра транзакции:

```
ASSIGN <№ параметра> [<суффикс>], <значение> [, <функция>]
```

№ параметра	—	номер параметра, связанного с транзакцией;
суффикс	—	операция (+ или -);
значение	—	присваиваемое/прибавляемое/вычитаемое значение;
функция	—	модификатор значения.

Следующий блок предназначен для сохранения текущего модельного времени в заданном параметре текущей транзакции.

```
MARK <№ параметра>
```

Позднее доступ к данной отметке у текущей транзакции может быть осуществлён при помощи выражения-системного числового атрибута `MP<№ параметра>` (см. пример на с. 77).

# Пример организации цикла для заявок

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

\* *Константы*

Counter EQU 13 ; Введение значения

\* *Блоки модели*

GENERATE 5,2,,100 ; Генерирование 100 заявок

ASSIGN Counter,5 ; Присвоение параметру значения

SEIZE Facility1 ; Захват канала

Loop1 ADVANCE 7,2 ; Задержка обслуживания

LOOP Counter,Loop1 ; Цикл заявки

RELEASE Facility1 ; Освобождение канала

TERMINATE 1 ; Выход заявки

\* *Команды*

START 100



- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Циклы
- Параметры
- Пример 9
- Отчёт 9
- Сравнение
- Пример 10.1
- Пример 10.2
- Пояснения 10
- Состояния
- Расщепление
- Сборка
- Пример 11
- Отчёт 11
- Группирование
- Синхронизация
- Пример 12
- Отчёт 12
- Ситуации
- Заключение

NAME		VALUE							
COUNTER			13.000						
...									
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY				
LOOP1	1	GENERATE	100	0	0				
	2	ASSIGN	100	0	0				
	3	SEIZE	100	0	0				
	4	ADVANCE	500	0	0				
	5	LOOP	500	0	0				
	6	RELEASE	100	0	0				
	7	TERMINATE	100	0	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	100	0.998	34.859	1	0	0	0	0	0

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данный блок предназначен для перенаправления, либо задержки транзакции в зависимости от заданного условия.

```
TEST <отношение> <значение1>, <значение2> [, <метка>]
```

- |   |   |   |
|---|---|---|
| отношение                                     | — | E, G, GE, L, LE, или NE;  |
| значение <sub>1</sub> , значение <sub>2</sub> | — | сравниваемые значения;  |
| метка   | — | метка блока для перехода, если сравнение <i>ложно</i> (либо транзакция задерживается, если метка не указана). |

# Модель с двумя входящими/выходящими потоками заявок

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Рассмотрим систему с двумя входящими потоками заявок:

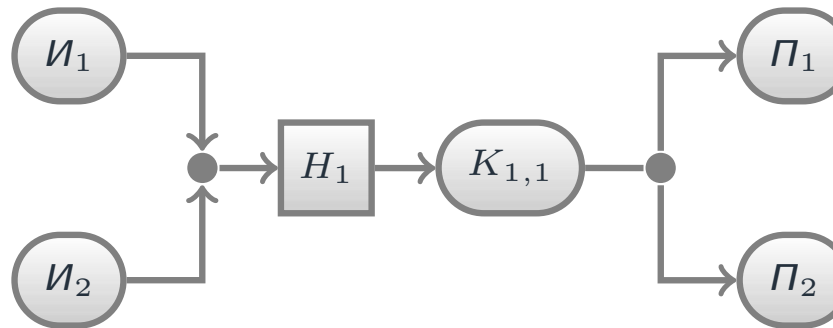


Рис. 5: Система массового обслуживания с двумя входящими/выходящими потоками заявок

Здесь заявки из обоих источников направляются в общий накопитель  $H_1$ , затем в одноканальное устройство обслуживания  $K_{1,1}$ . После обработки заявки направляются в приёмники  $P_1$  и  $P_2$ , причём в приёмник  $P_1$  должны направляться заявки, поступившие в систему из источника  $I_1$ , а в  $P_2$  — поступившие из  $I_2$ .

Эту модель можно описать при помощи программы (см. далее).

# Пример перенаправления потоков заявок

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

\* Блоки модели

\* I поток заявок

GENERATE 10 ; Создание транзакции

ASSIGN 13,1 ; Пометка I потока

TRANSFER ,Process ; Переход к обработке

\* II поток заявок

GENERATE 5 ; Создание транзакции

ASSIGN 13,2 ; Пометка II потока

\* Общее устройство обслуживания для обоих потоков

Process SEIZE Facility1 ; Вход в устройство

ADVANCE 7 ; Задержка

RELEASE Facility1 ; Выход из устройства

TEST E P13,1,Second ; Разделение потоков

\* Выход для I потока заявок

TERMINATE 1 ; Выход для I потока

\* Выход для II потока заявок

Second TERMINATE 1 ; Выход для II потока

\* Команды

START 100

# Пояснения к примеру моделирования очереди ограниченной ёмкости

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

- В данном примере потоки транзакций создаются двумя блоками **GENERATE**.
- После выхода из блоков **GENERATE** транзакции попадают в блоки **ASSIGN**, в которых их 13-му параметру присваивается значение 1 или 2 — в зависимости от номера потока. В дальнейшем эти отметки будут использоваться для определения принадлежности заявок входящим потокам.
- После выхода из блока **ASSIGN** транзакции второго потока попадают непосредственно в блок **SEIZE** с меткой `Process`, с которым связана очередь транзакций. Транзакции первого потока направляются в этот блок при помощи блока безусловного перехода **TRANSFER**.
- После завершения обработки на устройстве (блок **RELEASE**) заявки переходят в блок **TEST**, который направляет их либо на следующий блок (**TERMINATE**), либо по метке `Second` (второй блок **TERMINATE**).
- Условием, проверяемым блоком **TEST**, является равенство (**E**) значений 13-го параметра транзакции, занимающей блок в настоящий момент (выражение `p13`) и константы 1.

# Блок проверки состояния устройства

Следующий блок предназначен для перенаправления, либо задержки транзакции в зависимости от состояния заданного блока модели.

```
GATE <условие> <устройство> [, <метка>]
```

условие	—	заданное условие состояния устройства (см. далее);
устройство	—	имя заданного одноканального/многоканального устройства, блока <code>match</code> , либо логического ключа;
метка	—	метка блока для перехода, если условие <i>ложно</i> (либо транзакция задерживается, если метка не указана).

Условие	Значение
<code>FV, FNV</code>	Устройство доступно/недоступно
<code>I, NI</code>	Устройство прервано/не прервано
<code>U, NU</code>	Устройство используется/не используется
<code>SE, SNE</code>	Многоканальное устройство пусто/не пусто
<code>SF, SNF</code>	Многоканальное устройство заполнено/не заполнено
<code>SV, SNV</code>	Многоканальное устройство доступно/недоступно
<code>M, NM</code>	Блок <code>match</code> ожидает/не ожидает транзакции
<code>LS, LR</code>	Логический ключ находится в установленном/сброшенном состоянии

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данный блок для каждой проходящей через него транзакции создаёт заданное количество её копий, которые выходят из него вместе с оригинальной транзакцией. Все транзакции помечаются одним номером *семейства*, позволяющим установить соотношение копий оригиналу. При помощи данного блока можно моделировать расщепление заявки на части, либо одновременное поступление группы заявок.

```
SPLIT <количество>, [<метка>][, <№ параметра>]
```

количество	—	количество создаваемых копий;
метка	—	метка блока для перенаправления копий;
№ параметра	—	номер параметра транзакции, получающий номер копии.

*Замечание:* транзакции, создаваемой в блоке **GENERATE**, присваивается номер семейства, равный номеру транзакции. Транзакции-копии, создаваемой в блоке **SPLIT**, присваивается номер семейства транзакции-оригинала. △

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

**Сборка**

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данный блок уничтожает заданное количество проходящих через него копий транзакций для каждого семейства. При помощи данного блока можно моделировать сборку одной заявки (детали) из частей.

```
ASSEMBLE <количество>
```

количество — количество ожидаемых копий одного семейства (будет уничтожено копий на 1 меньше).



# Пример расщепления и сборки заявок

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

## \* Блоки модели

**GENERATE** 5,3,,100 ; Создание 100 транзакций  
**SPLIT** 1,Place1 ; Копирование транзакций на устройство № 1.2

## \* Устройство № 1.1

**SEIZE** Facility11 ; Захват устройства № 1.1  
**ADVANCE** 6,2 ; Задержка транзакции  
**RELEASE** Facility11 ; Освобождение устройства № 1.1  
**TRANSFER** ,Out1 ; Перенаправление транзакции на сборку

## \* Устройство № 1.2

Place1 **SEIZE** Facility12 ; Захват устройства № 1.2  
**ADVANCE** 9,1 ; Задержка транзакции  
**RELEASE** Facility12 ; Освобождение устройства № 1.2

## \* Сборка и выход

Out1 **ASSEMBLE** 2 ; Сборка 2 транзакций  
**TERMINATE** 1 ; Уничтожение транзакций

## \* Команды

**START** 100 ; Моделирование прохождения 100 транзакций

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Циклы
- Параметры
- Пример 9
- Отчёт 9
- Сравнение
- Пример 10.1
- Пример 10.2
- Пояснения 10
- Состояния
- Расщепление
- Сборка
- Пример 11
- Отчёт 11
- Группирование
- Синхронизация
- Пример 12
- Отчёт 12
- Ситуации
- Заключение

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES					
	0.000	903.304	11	2	0					
	...									
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY				
	1	GENERATE	100		0	0				
	2	SPLIT	100		0	0				
	3	SEIZE	100		0	0				
	4	ADVANCE	100		0	0				
	5	RELEASE	100		0	0				
	6	TRANSFER	100		0	0				
PLACE1	7	SEIZE	100		0	0				
	8	ADVANCE	100		0	0				
	9	RELEASE	100		0	0				
OUT1	10	ASSEMBLE	200		0	0				
	11	TERMINATE	100		0	0				
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
FACILITY11	100	0.659	5.956	1	0	0	0	0	0	
FACILITY12	100	0.992	8.963	1	0	0	0	0	0	

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

Синхронизация

Пример 12

Отчёт 12

Ситуации

Заключение

Данный блок осуществляет накопление заданного количества транзакций одного семейства с последующей попыткой одновременной передачи их в следующий блок. Отличие от блока **ASSEMBLE** заключается в том, что накапливаемые транзакции не уничтожаются.

**GATHER** *⟨количество⟩*

Следующий блок присваивает всем проходящим через него транзакциям заданный номер семейства:

**ADOPT** *⟨№ группы⟩*

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Циклы

Параметры

Пример 9

Отчёт 9

Сравнение

Пример 10.1

Пример 10.2

Пояснения 10

Состояния

Расщепление

Сборка

Пример 11

Отчёт 11

Группирование

**Синхронизация**

Пример 12

Отчёт 12

Ситуации

Заключение

Следующий блок осуществляет барьерную синхронизацию входящей в него транзакции: ожидание входа в заданный парный блок **MATCH** транзакции того же семейства с последующей попыткой передачи транзакции в следующий блок:

```
MATCH <метка>
```

метка — метка парного блока **MATCH**.

# Пример синхронизации транзакций

Пусть система массового обслуживания имеет структуру, представленную на рис. 6:

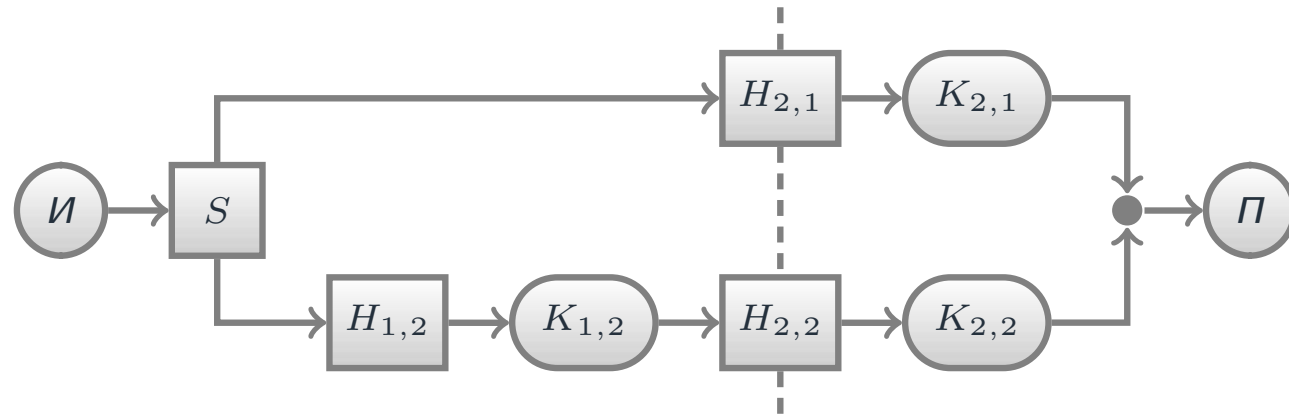


Рис. 6: Система массового обслуживания с барьерной синхронизацией заявок

Здесь входящая в систему заявка разделяется на две части в блоке  $S$ , после чего одна часть обрабатывается одним устройством ( $K_{2,1}$ ) с накопителем  $H_{2,1}$ , а вторая — последовательно двумя устройствами ( $K_{1,2}$ ,  $K_{2,2}$ , с соответствующими накопителями). При этом обработка обеих частей исходной заявки на устройствах  $K_{2,1}$  и  $K_{2,2}$  должна начинаться одновременно (для этого одна часть заявки должна ожидать в накопителе  $H_{2,1}$ ). Далее обе части выводятся из системы. Эту модель можно описать при помощи программы (см. далее).

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Циклы
- Параметры
- Пример 9
- Отчёт 9
- Сравнение
- Пример 10.1
- Пример 10.2
- Пояснения 10
- Состояния
- Расщепление
- Сборка
- Пример 11
- Отчёт 11
- Группирование
- Синхронизация
- Пример 12
- Отчёт 12
- Ситуации
- Заключение

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Циклы
- Параметры
- Пример 9
- Отчёт 9
- Сравнение
- Пример 10.1
- Пример 10.2
- Пояснения 10
- Состояния
- Расщепление
- Сборка
- Пример 11
- Отчёт 11
- Группирование
- Синхронизация
- Пример 12
- Отчёт 12
- Ситуации
- Заключение

```

* Блоки модели
    GENERATE 10,,100
    SPLIT 1,Place1
    SEIZE Facility11
    ADVANCE 5
    RELEASE Facility11
* Синхронизация со II потоком
Sync1 MATCH Sync2
    SEIZE Facility21
    ADVANCE 3
    RELEASE Facility21
* Перенаправление к выходу
TRANSFER ,Finish
    
```

```

Place1 SEIZE Facility12
        ADVANCE 10
        RELEASE Facility12
* Синхронизация с I потоком
Sync2 MATCH Sync1
        SEIZE Facility22
        ADVANCE 3
        RELEASE Facility22
* Выход из системы
Finish TERMINATE 1
* Команды
START 200
    
```

```

                START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
                0.000                1016.000  1013.000   18          4          0

                ...
FACILITY      ENTRIES  UTIL.    AVE. TIME AVAIL.  OWNER  PEND  INTER  RETRY
FACILITY11    100     0.494     5.000  1          0     0     0     0
FACILITY12    100     0.987    10.000  1          0     0     0     0
FACILITY21    100     0.592     6.000  1          0     0     0     0
FACILITY22    100     0.296     3.000  1          0     0     0     0
    
```

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

## Моделирование типичных ситуаций

# Моделирование работы в течение заданного времени

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

## \* Описание основных блоков модели

<b>GENERATE</b>	5,2	; Создание транзакции
<b>SEIZE</b>	Facility1	; Захват устройством
<b>ADVANCE</b>	12,5	; Задержка в устройстве
<b>RELEASE</b>	Facility1	; Выход из устройства
<b>TERMINATE</b>		; Счётчик завершения не уменьшается

## \* Обеспечение имитации в течение заданного времени

<b>GENERATE</b>	120	; Генерация через 120 мин.
<b>TERMINATE</b>	1	; Уничтожение транзакта

## \* Команды

<b>START</b>	1
--------------	---



# Пояснения к реализации моделирования в течение заданного времени

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

- Первый блок **TERMINATE** не имеет поля вычитаемого из счётчика моделирования (то есть, его значение по умолчанию равно 0). Переход произвольного количества заявок в этот блок не влияет на момент окончания моделирования.
- Второй блок **TERMINATE** определяет значение поля вычитаемого, равное 1. Находящийся непосредственно перед ним блок **GENERATE** сгенерирует заявку через 120 единиц модельного времени.
- Команда **START** определяет значение счётчика моделирования, равное 1. Это означает, что сгенерированная через 120 единиц времени заявка вторым блоком **GENERATE**, приведёт к завершению моделирования при попадании в блок **TERMINATE**.

# Моделирование пуассоновского потока событий

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

```
EXPONENTIAL(<№ генератора>, <t0>, <1/λ>)
```

Характеристики распределения:

$$p(t) = \begin{cases} \lambda e^{-\lambda(t-t_0)} & t \geq t_0 \\ 0 & t < t_0 \end{cases}, \quad M\tau = t_0 + \frac{1}{\lambda}, \quad D\tau = \frac{1}{\lambda^2}$$

Пример:

```
GENERATE (EXPONENTIAL(1,0.0,6.0))
```

# Динамическое изменение параметров блоков

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

```
Multi1 STORAGE 100 ; Ёмкость устройства = 100
* Блоки
    GENERATE 10 ; Создание транзакции
    ASSIGN 13,(10+RN1@5) ; 13-му параметру := (10 .. 14)
    ENTER Multi1,P13 ; Захват (10 .. 14) единиц
    ADVANCE 10 ; Задержка
    LEAVE Multi1,P13 ; Освобождение столько же единиц
    TERMINATE 1 ; Уничтожение транзакции
* Команды
    START 100
```

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Ограничение 1
- Ограничение 2
- Пуассон
- Изменение
- Отчёт 13**
- Время
- Отчёт 14
- Заключение

...									
LABEL	LOC	BLOCK TYPE			ENTRY COUNT	CURRENT	COUNT	RETRY	
	1	GENERATE			100		0	0	
	2	ASSIGN			100		0	0	
	3	ENTER			100		0	0	
	4	ADVANCE			100		0	0	
	5	LEAVE			100		0	0	
	6	TERMINATE			100		0	0	
...									
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY
MULTI1	100	100	0	28	1224	1	12.119	0.121	0
...									

# Проверка на прохождение участка модели за заданное время

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Ограничение 1

Ограничение 2

Пуассон

Изменение

Отчёт 13

Время

Отчёт 14

Заключение

\* Блоки

```
GENERATE 10 ; Создание транзакции
MARK 13 ; 13-му параметру := текущее время
SEIZE Facility1 ; Захват устройства
ADVANCE 9,3 ; Задержка
RELEASE Facility1 ; Освобождение устройства
```

\* Если разность текущего времени и времени, записанного в 13-м  
\* параметре меньше 10, перейти к следующему блоку. Иначе  
\* перейти по метке Lost

```
TEST L MP13,10,Lost ; Время обслуживания < 10?
```

\* Выход заявок, обработанных быстрее, чем за 10 единиц времени

```
TERMINATE 1 ; Уничтожение транзакции
```

\* Выход заявок, обработанных за 10 единиц времени и дольше

```
Lost TERMINATE 1 ; Уничтожение транзакции
```

\* Команды

```
START 100
```

- Концепции
- Основные блоки
- Перенаправление
- Недоступность
- Циклы
- Ситуации
- Ограничение 1
- Ограничение 2
- Пуассон
- Изменение
- Отчёт 13
- Время
- Отчёт 14
- Заключение

...									
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY			
	1	GENERATE	101	0	0				
	2	MARK	101	0	0				
	3	SEIZE	101	1	0				
	4	ADVANCE	100	0	0				
	5	RELEASE	100	0	0				
	6	TEST	100	0	0				
	7	TERMINATE	46	0	0				
LOST	8	TERMINATE	54	0	0				
...									
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
FACILITY1	101	0.897	9.001	1	101	0	0	0	0
...									

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Литература

## Заключение

Концепции

Основные блоки

Перенаправление

Недоступность

Циклы

Ситуации

Заключение

Литература

- [1] *Боев В. Д.* Моделирование систем. Инструментальные средства GPSS World. — СПб.: БХВ-Петербург, 2004. — 368 с.
- [2] *Кравченко П. П., Хусаинов Н. Ш.* Имитационное моделирование вычислительных систем средствами GPSS/PC. — Таганрог: Изд.-во ТРТУ, 2000. — 116 с.
- [3] *Кудрявцев Е. М.* GPSS World. Основы имитационного моделирования различных систем. — М.: ДМК Пресс, 2004. — 320 с.
- [4] *Советов Б. Я., Яковлев С. А.* Моделирование систем. Практикум: Учеб. пособие для вузов. — 3-е, стер. изд. — М.: Высш. шк., 2005. — 295 с.
- [5] *Шрайбер Т. Дж.* Моделирование на GPSS: Пер. с англ. — М.: Машиностроение, 1980. — 592 с.