

Тема 10.

Блоки, що забезпечують побудову моделі типу "СМО з одним пристроєм". Забезпечення пріоритетного обслуговування

10.1. Блоки, що забезпечують побудову моделі типу "СМО з одним пристроєм"

Потоки, які існують у реальних системах, у моделях імітують за допомогою транзактів. Тому, передусім, розглянемо введення транзактів у модель і виведення з неї.

Введення транзактів у модель

GENERATE (генерувати) — це блок, через який транзакти входять у модель. На кількість різних блоків GENERATE в одній моделі обмежень не існує. Інтервал часу між послідовними появами транзактів блоку GENERATE називають *інтервалом надходження*.

Коли транзакт входить у модель через блок GENERATE, планувальник планує час надходження наступного транзакта шляхом розіграшу випадкового числа відповідно до розподілу інтервалів часу надходження з подальшим додаванням розіграного значення до поточного значення таймера. При досягненні цього значення часу наступний транзакт вводиться в модель через блок GENERATE і т. д.

Інформація про ймовірнісний розподіл інтервалу надходження задається за допомогою операндів А і В. Усі можливі розподіли інтервалів часу надходження в GPSS класифікують на рівномірно розподілені й усі інші розподіли (детальніше про розподіли у підрозділі 12.3).

Блок GENERATE має такий формат:

GENERATE A,B,C,D,E

Операнд А задає середній інтервал часу між послідовними надходженнями транзактів у модель.

Операнд В задає половину поля допуску для значення операнда А. Операнд В називають модифікатором, він змінює значення інтервалу генерування транзактів порівняно з інтервалом, зазначеним операндом А. Є два типи модифікаторів: модифікатор-інтервал і модифікатор-функція.

За допомогою модифікатора-інтервала задають рівномірний закон розподілу часу між створенням транзактів.

Операнди А і В можуть бути іменем, додатним числом, виразом у дужках або безпосередньо СЧА.

Різниця А–В значень, заданих операндами А і В, дає нижню межу інтервалу, а сума А+В – його верхню межу. Після генерування чергового транзакта вибирається число з отриманого інтервалу. Це буде значенням часу, через який наступний транзакт вийде з блоку GENERATE.

Наприклад, блок

GENERATE 15,5

створює транзакти через випадкові інтервали часу, рівномірно розподілені на відрізьку [10; 20].

Операнди А і В задавати не обов'язково, їх значення за замовчуванням — "нуль".

Модифікатор-функцію використовують, якщо закон розподілу інтервалу надходження відрізняється від рівномірного. У цьому випадку у полі В записують посилання на функцію (її СЧА), яка описує потрібний закон розподілу. Під дією модифікатора-функції значення операнда А множиться на значення функції, заданої операндом В.

Задаючи операнди А і В слід пам'ятати, що за будь-якого способу обчислення інтервалу часу значення операнда В не повинне перевищувати значення операнда А.

Операнд С задає зміщення інтервалів (початкову затримку), тобто момент часу появи першого транзакта у блоці GENERATE. Після появи першого транзакта всі наступні появи транзактів виникають згідно з розподілом інтервалів часу, заданих операндами А і В. Операнд С можна використовувати як для прискорення, так і для уповільнення приходу першого транзакта або для призначення приходу в потрібний момент часу. Початкова затримка може бути меншою, такою самою або більшою порівняно із середнім часом, заданим операндом А. Коли операнд С не використовують, то інтервали генерування визначають операнди А і В (вони не впливають на затримку). Операнд С може бути іменем, додатним числом, виразом у дужках або безпосередньо СЧА.

Операнд D задає граничне значення загальної кількості транзактів, які можуть увійти в модель через цей блок GENERATE протягом часу моделювання. Коли ця кількість досягнута, даний блок GENERATE

перестає бути активним. Якщо не визначене граничне значення (операнд D не використовується), то блок GENERATE залишається активним до завершення моделювання.

Операнд E встановлює клас пріоритету кожного з транзактів, які потрапляють у модель через блок GENERATE. Для задавання пріоритетів з метою підвищення ефективності роботи GPSS World рекомендується використовувати послідовність цілих чисел 0, 1, 2, 3,... Чим більше число, тим вищий пріоритет. Якщо операнд E не використовується, то за замовчуванням пріоритет створених даним блоком GENERATE транзактів дорівнює нулю.

Операнди D та E можуть бути іменем, додатним числом, виразом у дужках або безпосередньо СЧА, але вони можуть мати значення лише цілих додатних і цілих чисел відповідно.

У будь-якому блоці GENERATE має бути заданий хоча б один з операндів A і D. ***Не можна використовувати на полі операнда параметри транзактів.***

У початковий момент часу в кожному блоці GENERATE здійснюється підготовка до виходу одного транзакта. На цій стадії модель ще цілковито не ініціалізована до виконання. З цієї причини всі СЧА, використані у блоці GENERATE, повинні бути вже визначені раніше. Тому в моделі перед блоком GENERATE мають стояти команди визначення EQU, INITIAL, FUNCTION, VARIABLE, FVARIABLE.

Приклади запису блоків GENERATE:

GENERATE	10
GENERATE	25,15
GENERATE	5,,30
GENERATE	6,3,20,100
GENERATE	15,,20,,2

У першому прикладі середній час появи транзактів у моделі складає 10 одиниць модельного часу. У другому прикладі час появи транзактів у моделі рівномірно розподілений на інтервалі [10; 40]. У третьому прикладі перший транзакт з'явиться в моделі в момент часу, рівний 30, надалі транзакти з'являтимуться в моделі кожні 5 одиниць модельного часу. У четвертому прикладі блоком GENERATE буде згенеровано 100 транзактів, причому перший з'явиться в моделі в момент часу рівний 20, а інтервали появи наступних транзактів рівномірно розподілені на інтервалі [3; 9]. У п'ятому прикладі починаючи з 20-ї одиниці модельного

часу кожні 15 одиниць часу в моделі з'являтимуться транзакти з пріоритетом, рівним 2.

Вилучення транзактів з моделі та завершення моделювання

Транзакти вилучаються з моделі, потрапляючи до блока TERMINATE (завершити). Блок TERMINATE завжди дає дозвіл на входження всім транзактам, які намагаються це зробити. У моделі може бути будь-яка кількість блоків TERMINATE. Блок має такий формат запису:

TERMINATE A

Значенням операнда A є кількість одиниць, на яку блок TERMINATE зменшує значення *лічильника завершень* під час надходження транзакта до цього блока. Операнд A може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. За замовчуванням значення операнда A дорівнює нулю. У цьому випадку транзакт знищується, а значення лічильника завершень не змінюється.

Лічильник завершень є коміркою пам'яті, яка зберігає додатне ціле число, записане на початку моделювання командою START. Поточне значення лічильника завершень доступне для користувача через системний СЧА TG1.

У процесі моделювання транзакти потрапляють до блока TERMINATE і віднімають число, яке дорівнює значенню операнда A, від лічильника завершень. У момент, коли значення лічильника досягає нуля, моделювання завершується.

Необхідно звернути увагу на той факт, що хоча в моделі може бути багато блоків TERMINATE, лічильник завершень тільки один.

Моделювання починається, коли зустрічається команда START (почати). Для визначення початкового значення лічильника завершення використовується операнд A команди START.

Розглянемо приклад, в якому блок TERMINATE і команда START використовуються для управління процесом моделювання.

Припустимо, що одиницею часу обрана 1 хвилина і необхідно промоделювати поведінку системи протягом 8 годин (480 хв.), потім моделювання має закінчитися. Це можна зробити включивши в модель сегмент з двох блоків:

GENERATE	480
TERMINATE	1

Цей сегмент з двох блоків забезпечує завершення моделювання у момент модельного часу, рівний 480.

У всіх інших блоках TERMINATE в моделі мається на увазі використання операнда A за замовчуванням (нуль). Це означає, що завершення моделювання, визначене лічильником завершень, не залежатиме від інших блоків TERMINATE.

У команді START як операнд A повинна бути використана одиниця:

```
START      1
```

Одиничне значення операнда A команди START викликає запис у лічильник завершень одиниці як початкове значення. У процесі моделювання завершення рухів транзактів, які відбуваються час від часу в інших блоках TERMINATE, не впливають на лічильник завершень. Отже, у момент модельного часу, що дорівнює 480, транзакт потрапить у блок GENERATE. Він відразу ж перейде в наступний блок TERMINATE. Оскільки операнд цього блоку є одиниця, то з лічильника завершень віднімається одиниця. Це зменшує значення лічильника від одиниці до нуля. У результаті інтерпретатор припиняє моделювання.

Тепер припустимо, що для виконання дій, описаних у попередньому прикладі, розробник вставляє в модель сегмент з двох блоків:

```
GENERATE    1
```

```
TERMINATE   1
```

(Інший вид сегменту з двох блоків, що спричиняє завершення моделювання у момент часу, рівний 480).

У всіх інших блоках TERMINATE моделі мається на увазі використання операнда A за умовчанням; як операнд A команди START використовується число 480:

```
START 480
```

Відмітимо, що у блоці GENERATE як інтервал часу задана одиниця. Інакше кажучи, транзакти з'являються тут в моменти часу 1, 2, 3, 4, ..., 478, 479 і 480. Кожен з цих транзактів потрапляє в наступний блок TERMINATE, де з лічильника завершень віднімається одиниця при видаленні транзакта. Коли у блок TERMINATE увійде 480-й транзакт, значення лічильника завершень вже буде дорівнювати одиниці. 480-й транзакт викликає зменшення його від одиниці до нуля і моделювання завершується.

Метод завершення моделювання, представлений у першому прикладі, є переважнішим, хоча в логічному сенсі вони рівнозначні, оскільки

останній приклад вимагає 480 виконань обробки блоків GENERATE і TERMINATE. Оскільки обробка кожного блоку вимагає часу EOM, то останній приклад принаймні в 480 разів дорожче попереднього. При другому способі завершення моделювання інтерпретатор GPSS пройде всі фіксовані моменти часу, оскільки таймер повинен приймати значення 1, 2, 3, 4 ..., 478, 479 і 480 через те, що транзакти входять у модель через блок GENERATE. Це значною мірою збільшує число приростів значень часу до таймера у процесі моделювання, що сприяє ще більшому збільшенню часу моделювання.

Користувач задає час моделювання, зазначаючи в команді START значення лічильника завершень.

Команду START використовують для запуску процесу моделювання. Вона має такий формат запису:

START A,B,C,D

Операнди:

A — задає значення лічильника завершень, яке визначає момент завершення прогону моделі, і може бути лише цілим додатним числом;

B — операнд виведення статистики. Він може бути NP (немає виведення даних) або відсутнім. У випадку задання NP стандартний звіт не виводиться. За умовчанням виводиться стандартний звіт;

C — не використовується і збережений для сумісності з описами попередніх версій GPSS. Його використовують як лічильник "кадрів" для періодичного виведення звіту. Ця функція доступна у випадку використання в моделі кількох команд START і REPORT;

D — визначає необхідність виведення вмісту списків подій. Якщо у полі операнда D зазначити будь-яке додатне ціле число, то списки поточних і майбутніх подій вводяться до стандартного звіту. Якщо поле операнда D порожнє, то за умовчанням списки у стандартному звіті не виводяться.

Команду START можна записувати в кінці програми моделі (тоді після трансляції моделі відразу починається моделювання) або можна вводити у програму моделі в інтерактивному режимі.

Інколи може виникнути необхідність завершити моделювання не через певний час, а після проходження через модель певної кількості транзактів, які імітують, наприклад, виготовлені деталі, передані каналом зв'язку повідомлення тощо. У цьому випадку сегмент задання часу моделювання не потрібен. Для організації такого способу завершення моделювання потрібно у блоках TERMINATE, які виводять з моделі транзакти, кількість яких визначає момент завершення моделювання, зазначити число, на яке зменшується лічильник

завершень. У команді START також вказують число, ділення якого на записане у блоці TERMINATE число дасть потрібну кількість транзактів, які мають пройти через модель. Наприклад, якщо потрібно завершити моделювання після проходження через модель 1000 транзактів, то вона має містити такий сегмент

```
GENERATE      100,40
```

```
.....
```

```
...
```

```
TERMINATE    1
```

```
START        1000
```

Блоків TERMINATE, які виводять з моделі транзакти, сумарна кількість яких, як у наведеному прикладі, дорівнює 1000, може бути кілька. Усі вони повинні мати 1 у полі операнда A. В інших блоках TERMINATE, якщо вони є в моделі, поле операнда A має бути порожнім.

Моделювання пристроїв

У GPSS елементами, які потребують обслуговування, є транзакти. Вони обслуговуються деякими пристроями: запити обробляються ЕОМ, робочі ремонтують обладнання, деталі обробляються верстатами тощо. Незалежно від того, люди це або предмети, обслуговуючі елементи називають пристроями чи приладами.

Пристрої характеризуються двома основними властивостями: кожен пристрій у будь-який момент часу може обслуговувати лише один транзакт. Якщо у процесі обслуговування з'являється новий транзакт, то він повинен або почекати в черзі на своє обслуговування, або залишити систему необслуженим, або перервати обслуговування, яке відбувається в даний момент;

коли на пристрій надходить транзакт, у моделі потрібно передбачити час, необхідний для його обслуговування.

Оскільки пристроїв у моделі може бути багато, то для їхньої ідентифікації їм присвоюють імена — числові або символічні.

У загальному випадку у процесі моделювання використання пристрою відбуваються такі події:

- 1) очікування своєї черги;
- 2) займання пристрою;
- 3) обслуговування пристроєм;

4) вивільнення пристрою.

Під час моделювання можливі такі режими організації функціонування пристроїв:

займання пристрою і його звільнення; переривання обслуговування пристроєм; недоступність пристрою і відновлення доступності. **Зайняття та звільнення пристрою**

Транзакти рухаються в моделі від блока до блока. Якщо в певний момент активності транзакт має намір зайняти пристрій, то для цього він заходить (або намагається увійти) до відповідного блока, який описує цей пристрій.

Блок, який описує пристрій, повинен мати такі властивості:

якщо пристрій вже використовують, то транзакт не може увійти до блока і мусить чекати у черзі або має бути вилучений з моделі;

якщо пристрій не використовують, то транзакт може увійти до блока. Унаслідок цього статус пристрою змінюється на "зайнятий".

Блок, що має такі властивості, це блок SEIZE (зайняти). Потрапляння транзакта у блок SEIZE моделює займання пристрою.

Після завершення обслуговування потрапляння того самого транзакта до блока RELEASE (звільнити) моделює звільнення пристрою і призводить до зміни стану пристрою із "зайнятий" на "незайнятий".

Формати блоків:

SEIZE A

RELEASE A

В обох блоках операнд A — це ім'я або номер відповідного пристрою, який займає (звільняє) транзакт. Операнд може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. У той час як транзакти перебувають у моделі тимчасово, пристрої, які використовують у моделі, існують постійно протягом всього процесу моделювання.

Якщо транзакт заходить у блок SEIZE, то пристрій, зазначений у полі операнда A, перейшовши у стан "зайнятий", залишається в цьому стані аж до того часу, поки той самий транзакт не пройде через відповідний блок RELEASE, звільняючи пристрій.

Перед тим, як звільнити пристрій, транзакт може пройти через необмежену кількість блоків. Наприклад:

SEIZE Prystriy

RELEASE Prystriy

Якщо пристрій на ім'я Prystriy не зайнятий, то активний транзакт займає його. Якщо пристрій зайнятий, то транзакт переводиться до списку затримки цього пристрою позаду транзактів з тим самим пріоритетом і не заходить до блока SEIZE. Транзакту також не дозволяють увійти до блока SEIZE, якщо пристрій на ім'я Prystriy перебуває в недоступному стані.

Пристрій, як вже було зазначено, може мати ім'я або номер. У блоках SEIZE і RELEASE дозволяється записувати в полі операнда А номер безпосередньо без попереднього присвоєння його імені командою EQU. Наприклад:

SEIZE 4

RELEASE 4

Блоки SEIZE і RELEASE у разі виникнення необхідності можна перевизначати. Для цього вони повинні мати мітки.

Переривання обслуговування пристроєм

Переривання початого обслуговування можливе тільки для пристроїв і практикується лише при винятковій важливості транзакта, який щойно прийшов. Типовими прикладами є аварійні ситуації, потреба в екстренній медичній або іншій допомозі. Допущення переривань посилює ефект обслуговування з відносним пріоритетом.

Переривання і подальше поновлення початого обслуговування організовується блоками PREEMPT і RETURN. Заявка (транзакт), яка є причиною переривання, сама не може бути перерваною. Ім'я пристрою вказується в полі А блоку PREEMPT. Перерваний транзакт із записаним залишком часу обслуговування поміщається у список переривань і повертається у список майбутніх подій, коли переривання закінчиться.

Блок RETURN знімає переривання із вказаного в полі А пристрою. Якщо у блоці PREEMPT будуть транзакти вищого пріоритету, ніж перерваний, пристрій займе один з них.

PREEMPT A,B,C,D,E

Операнди:

А – ім'я або номер пристрою, що переривається;

В – режим зайняття пристрою. Операнд (необов'язковий) має бути PR або Null;

С — блок нового призначення транзакта, який у цей час володіє пристроєм;

D — номер параметра перерваного транзакта, який запам'ятовує недоданий час обслуговування, якщо перерваний транзакт видаляється зі списку майбутніх подій;

E — при значенні RE виключає перерваний транзакт з боротьби за пристрій (тоді операнд C необов'язковий).

Якщо пристрій не зайнятий, то блок PREEMPT поводитьсь подібно SEIZE. Інакше залежно від операнда B він діє в режимі пріоритету або в режимі переривання. У режимі пріоритету тільки транзакт з вищим пріоритетом може витіснити інший транзакт з пристроєм. У режимі переривання (порожнє поле B), якщо пристрій вже зайнятий по пріоритету, активний транзакт поміщається у список завислих транзактів. При наданні пристрою транзактам із цього списку віддається більша перевага, ніж тим, що йдуть по пріоритету або зі списку затримки.

Операнди C — E визначають, що потрібно зробити з перерваним транзактом. Таким транзактам не дозволяється бути у списку поточних подій. Проте зняті з пристрою транзакти, перебування яких у блоці ADVANCE не завершено, поміщаються у список поточних подій автоматично, якщо у блоці PREEMPT опущені операнди C і E.

Транзакту не дозволяється входити у блок PREEMPT в режимі переривання, якщо пристрій у поточний момент зайнятий по пріоритету. Таким чином, GPSS не забезпечує імітацію вкладених переривань.

Блок RETURN A звільняє пристрій A або видаляє перерваний транзакт з боротьби за пристрій.

Тут можуть бути два випадки. Якщо транзакт у поточний момент володіє пристроєм, то він звільняє пристрій і намагається увійти до наступного блоку. Якщо є потреба вирішити питання про передачу пристрою наступному транзакту претендент послідовно розшукується у списках завислих, перерваних і, нарешті, затриманих транзактів. У другому випадку транзакт відстороняється від пристрою по пріоритету. Тут транзакт видаляється зі списку перерваних транзактів для цього пристрою, причому його право на володіння пристроєм не зачіпається.

Окремої розмови заслуговує питання про дообслуговування перерваної заявки. Для реалізації цієї дисципліни використовується операнд D блоку PREEMPT — обчислюється час, що залишився, і заноситься в параметр перерваного транзакта. При повторному перегляді списку поточних подій перерваний транзакт входить у блок ASSIGN, що працює в режимі приросту. Там до нього додається додаткова затримка —

фіксована або випадкова. При відновленні в блоці ADVANCE вміст згаданого параметра визначить затримку, що фіксується у списку майбутніх подій.

Недоступність пристрою і відновлення доступності

Пристрої, окрім ступеня зайнятості, характеризуються поточною готовністю. Режим готовності пристрою вводиться блоком FAVAIL, режим неготовності – блоком FUNAVAIL. Ця властивість дозволяє моделювати поломки техніки, регламентні роботи, перерви у касирів на здачу грошей і продавців — на прийом товару, роботу з перервами на обід, тижневий і річний ритми. Часи поломок генеруються аналогічно моментам прибуття, інтервали між ними зазвичай підпорядковані розподілу Вейбулла.

FAVAIL A

Блок встановлює вказаний операндом A пристрій у стан готовності.

FUNAVAIL A,B,C,D,E,F,G,H

Блок переводить пристрій у стан неготовності для вхідних транзактів.

Операнди:

A – ім'я або номер пристрою;

B — розпорядження подальших дій для транзакта, що займав пристрій. Операнд необов'язковий, може бути RE, CO (REmove — видалити, COntinue – продовжити) або Null;

C – наступний блок для транзакта, що займав пристрій;

D — номер параметра для реєстрації часу, що залишився до закінчення обслуговування, якщо транзакт, що займав пристрій, видаляється зі списку майбутніх подій;

E – розпорядження подальших дій для транзактів, обслуговування яких було перерване до виконання блока FUNAVAIL. Операнд має бути RE, CO або Null;

F — новий блок для транзактів, виконання яких було перерване раніше;

G — подальші дії для транзактів, що чекали можливості зайняти пристрій до виконання блока FUNAVAIL. Операнд має бути RE, CO або Null;

H — новий блок, до якого перейдуть транзакти, що чекали можливості зайняти пристрій до виконання блока FUNAVAIL.

Складність блоку FUNAVAIL пояснюється тим, що він має справу з трьома типами транзактів:

- 1) тими, що займають пристрій (операнди B — D);
- 2) тими, що очікують поновлення обслуговування, перерваного блоком FUNAVAIL (операнди E — F);
- 3) тими, що очікували можливості зайняти пристрій до виконання FUNAVAIL (операнди G — H).

Транзакти, що надійшли в період неготовності, будуть затримані і не допущені до пристрою. Якщо в операндах записана опція RE, транзакти перестають претендувати на зайняття пристрою. Якщо в полі G для транзактів, що чекали можливості зайняти пристрій, вказано RE, то для подальшого перенаправлення транзактів треба заповнити поле H. Якщо використовується опція CO, то транзакти, що займали пристрій, можуть продовжувати його займати навіть у період неготовності. У статистичних даних з використання пристрою цей час вразовуватиметься.

Якщо вказується блок нового призначення, транзакти звільняються від раніше встановлених зв'язків і прямують до нового блока. Затримані і перервані транзакти, керовані операндами G і H, не можуть бути перенаправлені без опції RE. Транзакт, що зайняв пристрій і керований операндами B — D, а також перервані транзакти, керовані операндами E і F, можуть знову претендувати на заняття пристрою, навіть будучи направленими за новим призначенням. Це робиться вказівкою зміненого призначення без використання опції RE.

Операнди B — D використовуються для управління транзактом, що володів пристроєм. Якщо B = CO, цей транзакт не позбавляється володіння пристроєм, але може отримати нове призначення через операнд C. При B = RE він видаляється і може продовжувати шлях у моделі, не заходивши в RETURN або RELEASE для даного пристрою. Тут операнд C обов'язковий.

Реалізація затримки на час обслуговування

Припустимо, що транзакт тільки що увійшов у блок SEIZE, тобто зайняв пристрій. Після виконання відповідної підпрограми для даного блоку планувальник відразу ж намагається просунути транзакт у наступний послідовний блок моделі. Існує мало обмежень на те, яким повинен бути цей блок. Це, наприклад, може бути інший блок SEIZE, що описує інший пристрій. Так робиться, якщо необхідно, наприклад, виконати

операцію з одночасним зайняттям і робітника, і якого-небудь інструменту для виконання певного виду обслуговування. Якщо робітник й інструмент моделюються деякою парою пристроїв, то транзакт одночасно повинен зайняти обидва ці пристрої, перш ніж може початися обслуговування.

Зазвичай, транзакт займає пристрій для того, щоб негайно почати в ньому своє обслуговування, яке триває деякий час. Протягом цього часу транзакт повинен припинити рух у моделі. Лише після завершення обслуговування він має потрапити у блок RELEASE і звільнити пристрій.

Для затримки транзакта на певний відрізок модельного часу використовують блок ADVANCE. Найчастіше цей проміжок задають випадковою змінною. Формат запису блоку ADVANCE:

ADVANCE A, B

Операнди:

A – задає середній час обслуговування;

B – задає половину поля допуску для значення операнда A.

Операнди A і B мають такий самий сенс, як і відповідні операнди блока GENERATE. Кожен з операндів A і B може бути іменем, числом, виразом у дужках, СЧА або СЧА*параметром.

Як і у блоці GENERATE, модифікатори можуть бути двох типів: модифікатор-інтервал і модифікатор-функція, отже, блок ADVANCE підраховує час затримки (приросту модельного часу) такими самими способами.

Якщо задані операнди A і B, і B не визначає функцію, то обидва операнди підраховуються (якщо вони не константи) і за час затримки береться значення випадкової величини, рівномірно розподіленої на проміжку [A–B, A+B]. Для розігрування випадкових чисел можна вибрати будь-який генератор рівномірно розподілених випадкових чисел (задається на сторінці Random Numbers у журналі Settings моделі в полі блока ADVANCE).

Приклади:

ADVANCE 15

ADVANCE 60,30

У першому прикладі обслуговування транзакта триватиме 15 одиниць модельного часу. У другому прикладі тривалість обслуговування транзакта рівномірно розподілена на інтервалі від 30 до 90 одиниць модельного часу.

Так само, як і у блоці GENERATE, незважаючи на спосіб обчислення часу затримки, значення операнда В не повинне перевищувати значення операнда А.

Для задання часу затримки, розподіленого за законом, відмінним від рівномірного, можна використовувати модифікатор-функцію, яку записують у полі операнда В. Під час звертання до функції планувальник визначає деяке число — значення функції, результат множення якого на значення операнда А використовується як час затримки.

Блок ADVANCE ніколи не перешкоджає потраплянню до нього транзакта. Будь-яка кількість транзактів може перебувати в цьому блоці одночасно. Блок ADVANCE можна розташовувати в будь-яких місцях моделі, а не лише між блоками SEIZE і RELEASE. Час затримки може дорівнювати нулю. У такому випадку транзакт у блоці ADVANCE не затримується і переходить до наступного блока.

Лише блоки GENERATE і ADVANCE дають змогу розмістити транзакти у списку майбутніх подій. За допомогою цих блоків моделюють тривалість певної події або проміжок часу між настанням якихось подій.

10.2. Забезпечення пріоритетного обслуговування

Те, як впливає рівень пріоритетів транзактів на обробку, видно при розгляді списку поточних подій. При перенесенні транзактів зі списку майбутніх подій у список поточних подій транзакти у списку поточних подій займають положення, що є останнім усередині відповідного пріоритетного класу. Більш того, чим вище пріоритетний клас, тим ближче до початку списку поточних подій розташовується транзакт.

Оскільки планувальник GPSS проглядає список поточних подій з початку, він намагається просувати спочатку високопріоритетні транзакти. Пізніше, при перегляді він намагається просувати і транзакти, які мають низькі пріоритети.

Необхідність надання деяким транзактам пріоритетів часто витікає вже з постановки задачі.

Наприклад. При моделюванні одноколіїної ділянки залізниці з роз'їздом посередині пріоритет може бути наданий одному з напрямів, пасажирським поїздам перед товарними і т. п. На автодорожніх перехрестях пріоритет визначається відомою водіям системою правил і загальною ситуацією. При моделюванні роботи технічних систем аварії мають пріоритет вище ніж основні транзакти.

Транзакти можуть мати цілочисельні пріоритети (за замовчуванням — нульовий). Пріоритет призначається при генерації (операнд E блоку GENERATE) або блоком PRIORITY в довільному місці моделі.

Блок PRIORITY присвоює транзакту потрібний пріоритет. Пріоритет впливає на порядок опрацювання транзактів процедурою перегляду і порядок займання ними апаратних об'єктів.

Блок PRIORITY має такий формат запису:

PRIORITY A,B

Операнд A задає нове значення пріоритету і може бути іменем, цілим числом, СЧА або СЧА*параметром. Нове значення пріоритету може бути меншим, більшим або дорівнювати поточному значенню пріоритету. Поле B визначає режим BUFFER, у ньому вказується значення "BU".

Загалом процедура перегляду списку поточних подій намагається просунути активний транзакт до наступного блока. Однак, якщо в полі B блока PRIORITY записано слово BU, то після присвоєння транзакту нового значення пріоритету блок PRIORITY стає еквівалентним блоку BUFFER. Транзакт вводиться до списку поточних подій у кінець свого нового пріоритетного класу, процедура перегляду повертається до початку списку поточних подій і починає перегляд знову. Оскільки блок BUFFER не затримує транзакти, які увійшли в цей блок, то транзакт буде опрацьований процедурою перегляду знову без зміни значення модельного часу.

Наприклад:

PRIORITY 7

Тут активному транзакту присвоюється пріоритет, який дорівнює 7. Блок PRIORITY призначає новий рівень пріоритету і потім пересуває транзакт у списку поточних подій так, щоб він став останнім серед транзактів з відповідним значенням пріоритету. Ця операція виконується і в тому випадку, коли нове значення пріоритету дорівнює старому. Бувають випадки, коли потрібно змінити розташування транзакта в тому самому класі пріоритетів, в якому він перебуває.

Після того як планувальник завершить просування транзакта, який увійшов до блока PRIORITY, він автоматично повертається до початку перегляду списку поточних подій. Повернення до початку списку необхідне, оскільки у випадку, коли блок PRIORITY зменшив пріоритет активного транзакта, то планувальник пропустить усі транзакти, які перебувають у

списку поточних подій, починаючи з того місця, де був розташований активний транзакт до входження у блок PRIORITY, і закінчуючи його новим місцем у списку. Повернення до початку списку забезпечує перегляд усіх пропущених транзактів у той самий момент модельного часу.

Старшинство у пріоритеті само по собі не спричиняє переривання початого обслуговування – реалізується так званий відносний пріоритет, що стосується тільки очікування. Організація пріоритетного обслуговування з перериванням програмується спеціально. Інші дисципліни можуть бути запрограмовані роботою зі списком майбутніх подій. Вводити пріоритети має сенс тільки при великому завантаженні і малій питомій вазі пріоритетних заявок. У цих умовах він диференціює показники обслуговування – суттєво покращує їх для високопріоритетних заявок за рахунок деякого погіршення для низькопріоритетних. Інше призначення пріоритетів — модифікація логіки моделі.

Пріоритет і часові вузли

Призначення пріоритетів може потребуватися для забезпечення правильної послідовності обробки квазіодночасних подій ("часових вузлів"):

1. При одночасному прибутті заявки і закінченні обслуговування спочатку повинне завершитися обслуговування. Підвищення пріоритету обслуговуваного транзакта перед входом у блок ADVANCE дозволить запобігти необґрунтованій відмові у прийомі заявки. Так відбувається, наприклад, при вході і виході пасажирів автобуса — спочатку забезпечується вихід. Ще один часовий вузол утворюють вхід останнього пасажирів і відправлення автобуса – пасажир повинен встигнути увійти.

2. При одночасному надходженні машини на автозаправну станцію і закриттям станції в кінці робочого дня машина повинна бути обслужена. Для цього транзакту-машині при генерації повинен бути привласнений вищий пріоритет, ніж часовому транзакту.

Висновки

1. Кожному транзакту в досліджуваній системі у відповідність ставиться сегмент моделі, тобто самотійна послідовність зв'язаних блоків, яка описує весь життєвий цикл тимчасових елементів одного класу, починаючи з їх появи в системі і закінчуючи їх виходом з неї. Модель може включати декілька сегментів.

2. Кожен сегмент моделі починається з одного або декількох блоків GENERATE, які створюють транзакти, і закінчується блоком TERMINATE, який ці транзакти видаляє з моделі. Кожен транзакт, що з'явився в моделі, послідовно обробляється блоками, які входять у сегмент. Одночасно в моделі обробляється тільки один транзакт, решта транзактів моделі чекає своєї черги на обробку.
3. Переривання початого обслуговування практикується лише при винятковій важливості транзакта.
4. Пріоритет транзактів встановлюється при їх появі в моделі і може бути змінений у будь-якому її місці. Старшинство у пріоритеті впливає тільки на порядок обслуговування транзактів у моделі і не викликає переривання.

Контрольні запитання та завдання

1. Дайте визначення транзакта, обслуговуючого пристрою, черги.
2. Дайте визначення черги, обслуговування.
3. Назвіть блоки, що забезпечують побудову моделі типу "СМО з одним пристроєм". Особливості їх роботи.
4. Дайте визначення дисципліни обслуговування.
5. Яким чином у GPSS World забезпечується пріоритетне обслуговування?
6. Дайте визначення переривання обслуговування.
7. Яким чином у GPSS World організуються переривання обслуговування?