

Прикарпатський національний університет
імені Василя Стефаника

Махней О. В.

**Лабораторний практикум
з імітаційного моделювання у GPSS**

Частина 2

Методичні рекомендації
до проведення лабораторних занять

Івано-Франківськ
2012

УДК 004.94:519.682.6

ББК 32.973.26-018.2

МЗ6

Рекомендовано до друку Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника (протокол № 11 від 11 травня 2012 р.).

Рецензенти:

Федорук П. І., доктор технічних наук, професор (Прикарпатський національний університет імені Василя Стефаника),
Василишин П. Б., кандидат фізико-математичних наук, доцент (Прикарпатський національний університет імені Василя Стефаника)

МЗ6 Махней О. В. Лабораторний практикум з імітаційного моделювання у GPSS. Ч. 2 : методичні рекомендації до проведення лабораторних занять / Махней О. В. – Івано-Франківськ : Голіней, 2012. – 32 с.

Наведено методичні рекомендації до виконання лабораторних робіт на мові імітаційного моделювання GPSS. Призначено для проведення лабораторних занять з курсу «Методи комп'ютерної імітації».

Для студентів напряму підготовки «прикладна математика». Може бути корисним для студентів галузі знань «системні науки та кібернетика».

Зміст

Передмова	4
Лабораторна робота № 1. Вихід з ладу одноканального пристрою	5
Лабораторна робота № 2. Вихід з ладу одного каналу багатоканального пристрою	5
Лабораторна робота № 3. Вихід з ладу багатоканального пристрою	6
Лабораторна робота № 4. Моделювання роботи пристрою з чергою LIFO	7
Лабораторна робота № 5. Моделювання роботи пристрою зі спеціальною чергою обмеженої довжини	8
Лабораторна робота № 6. Неперервне моделювання	8
Лабораторна робота № 7. PLUS-процедури (двовимірний логнормальний закон розподілу)	10
Лабораторна робота № 8. Робота з матрицями	11
Найважливіші системні числові атрибути	11
Приклад імітації виходу з ладу одного каналу	13
Приклад імітації виходу з ладу багатоканального пристрою	15
Приклад створення спеціальної черги	18
Приклади інтегрування звичайних диференціальних рівнянь	21
PLUS-процедури	24
Приклади використання PLUS-процедур	26
Матриці	28
Налагодження моделі	29
Список рекомендованої літератури	32

Передмова

Моделювання – найбільш потужний універсальний метод дослідження і оцінювання ефективності різноманітних систем. Для моделювання складних систем, поведінка яких залежить від випадкових чинників, зокрема систем масового обслуговування, звичайно використовують імітаційне моделювання, бо недоліками більшості аналітичних моделей, побудованих на основі теорії масового обслуговування, є використання в них значних спрощень, таких, як розгляд вхідного потоку вимог як пуассонівського, припущення про експоненціальний розподіл часу обслуговування тощо. У той же час імітаційне моделювання знімає більшість такого роду обмежень і дозволяє створювати моделі як завгодно близькі до реальних систем.

У наш час, з розвитком комп'ютерної техніки значно зросли можливості імітаційного моделювання систем. Для створення моделей, звичайно, можна використовувати класичні алгоритмічні мови програмування, але простіше й ефективніше здійснювати моделювання за допомогою спеціалізованих мов імітаційного моделювання. Мова GPSS (General Purpose Simulation System – система моделювання загального призначення) була однією з найперших мов моделювання, а її сучасне середовище GPSS World, розроблене у 2000 році компанією Minuteman Software, є одним з найпотужніших і може використовуватись для професійного моделювання як дискретних, так і неперервних процесів у найрізноманітніших системах.

Методичні рекомендації містять завдання і методичні вказівки до лабораторних робіт, призначених для проведення лабораторних занять з використанням середовища GPSS World у межах курсу «Методи комп'ютерної імітації». На вивчення курсу відводиться 10 годин лекцій і 14 годин лабораторних занять. Використовується вільна версія програми, призначена для навчання студентів. Цей практикум є продовженням лабораторного практикуму [4].

Лабораторна робота № 1.

Вихід з ладу одноканального пристрою

Вхідний потік вимог має логнормальний закон розподілу з середнім значенням 2 і середньоквадратичним відхиленням 1 відповідного нормального розподілу. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною, розподіленою за експоненціальним законом з інтенсивністю 0,5. Вихід з ладу пристрою відбувається через інтервали часу, що мають розподіл Вейбулла з параметрами $\beta = 600$ і $\alpha = 10$ (задаються саме у цьому порядку у GPSS). Час ремонту є рівномірно розподіленою величиною на проміжку $[10; 100]$. При виході з ладу пристрою недообслужені вимоги чекають завершення дообслуговування після закінчення ремонту. За наявності черги першими мають обслуговуватись вимоги, обслуговування яких було перерване виходом з ладу пристрою.

1. Скласти модель на мові GPSS для моделювання роботи пристрою і його ремонту. Здійснити моделювання обслуговування 1000 вимог. Визначити середню і максимальну довжину черги. Проаналізувати таблицю часу перебування вимог у моделі.

2. Створити точку зупинки моделювання при виході з ладу пристрою. Визначити системний час, коли відбулась ця подія. Переглянути списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконати моделювання і проаналізувати модель у вікні блоків.

3. Створити і переглянути графік залежності поточної довжини черги від часу.

Лабораторна робота № 2.

Вихід з ладу одного каналу багатоканального пристрою

Вхідний потік вимог є пуассонівським з середнім значенням 0,3. Час обслуговування вимоги в одному каналі чотирика-

нального пристрою є випадковою величиною з логнормальним законом розподілу з середнім значенням 0,1 і середньоквадратичним відхиленням 0,5 відповідного нормального розподілу. Вихід з ладу одного з каналів пристрою відбувається через інтервали часу, що мають логнормальний розподіл з середнім значенням 6 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Час ремонту є рівномірно розподіленою випадковою величиною на проміжку [4; 30]. При виході з ладу каналу пристрою недообслужені вимоги залишають систему.

1. Скласти модель на мові GPSS для моделювання роботи пристрою і його ремонту. Здійснити моделювання обслуговування 10000 вимог. Визначити середню і максимальну довжину черги. Проаналізувати таблицю часу перебування вимог у моделі.

2. Під час моделювання переглянути зміну інформації у вікнах блоків, пристроїв, черг, таблиці. У вікні виразів під час моделювання переглянути зміну абсолютного системного часу, тривалості обслуговування вимоги, поточної довжини черги.

3. Створити точку зупинки моделювання при виході з ладу пристрою. Визначити системний час, коли відбулась ця подія. Переглянути списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконати моделювання і проаналізувати модель у вікні блоків.

4. Створити і переглянути графік залежності поточної довжини черги від часу.

Лабораторна робота № 3.

Вихід з ладу багатоканального пристрою

Вхідний потік вимог є пуассонівським з середнім значенням 5. Час обслуговування вимоги в одному каналі дванадцятиканального пристрою є випадковою величиною з логнормальним законом розподілу з середнім значенням 2 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Ви-

хід з ладу всього пристрою відбувається через інтервали часу, що мають логнормальний розподіл з середнім значенням 5 і середньоквадратичним відхиленням 3 відповідного нормального розподілу. Час ремонту є рівномірно розподіленою величиною на проміжку [50; 300]. При виході з ладу пристрою недообслужені вимоги чекають завершення ремонту. За наявності черги першими мають обслуговуватись вимоги, обслуговування яких було перерване виходом з ладу пристрою.

1. Скласти модель на мові GPSS для моделювання роботи пристрою і його ремонту. Здійснити моделювання обслуговування 1000 вимог. Визначити середню і максимальну довжину черги. Проаналізувати таблицю часу перебування вимог у моделі.

2. Під час моделювання переглянути зміну інформації у вікнах блоків, пристроїв, таблиці.

3. Створити точку зупинки моделювання при виході з ладу пристрою. Визначити системний час, коли відбулась ця подія. Переглянути списки поточних і майбутніх подій після досягнення цієї точки. Після цього по крокам виконати моделювання і проаналізувати модель у вікні блоків.

4. Створити і переглянути графік залежності поточної довжини черги від часу.

Лабораторна робота № 4.

Моделювання роботи пристрою з чергою LIFO

Вхідний потік вимог є пуассонівським з середнім значенням 5. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною з рівномірним законом розподілу на проміжку [2; 7]. За наявності черги першими мають обслуговуватись вимоги, які прийшли останніми.

1. Скласти модель на мові GPSS для моделювання роботи пристрою. Здійснити моделювання обслуговування 1000 вимог. Визначити середню і максимальну довжину черги, середній час

перебування в черзі. Проаналізувати таблицю часу перебування вимог у моделі.

2. Під час моделювання переглянути зміну інформації у вікнах блоків, пристроїв, таблиці.

3. Створити і переглянути графік залежності поточної довжини черги від часу.

Лабораторна робота № 5. Моделювання роботи пристрою зі спеціальною чергою обмеженої довжини

Вхідний потік вимог є пуассонівським з інтенсивністю 1. Час обслуговування вимоги в одноканальному пристрої є випадковою величиною з рівномірним законом розподілу на проміжку $[0,3; 1,5]$. За наявності черги першими мають обслуговуватись вимоги з найменшим часом обслуговування. Черга обмежена десятьма позиціями. Вимоги, для яких немає місця в черзі, залишають модель необслуженими.

1. Скласти модель на мові GPSS для моделювання роботи пристрою. Здійснити моделювання обслуговування 1000 вимог. Визначити середню і максимальну довжину черги, середній час перебування в черзі. Проаналізувати таблицю часу перебування вимог у моделі.

2. Під час моделювання переглянути зміну інформації у вікнах блоків, пристроїв, таблиці.

3. Створити і переглянути графік залежності поточної довжини черги від часу.

Лабораторна робота № 6. Неперервне моделювання

1. За допомогою GPSS знайти розв'язок початкової задачі $y' = 5 + 4\sqrt{y}$, $y(0) = 22$ у точці $x_1 = 33$. За допомогою GPSS

знайти таке значення x_2 , щоб розв'язок цієї початкової задачі $y(x_2) = 100$.

2. Скласти на мові GPSS модель п'ятиканальної системи масового обслуговування з відмовами, передбачивши можливість числового інтегрування системи диференціальних рівнянь для ймовірностей станів і порівняння результатів, отриманих за допомогою аналітичної та імітаційної моделей. Інтенсивність пуассонівського потоку замовлень – $\lambda = 1/3$, час обслуговування – випадкова величина з експоненціальним законом розподілу й інтенсивністю $\mu = 1/16$. Порівняти результати.

Вказівка. Система диференціальних рівнянь Колмогорова для ймовірностей станів має вигляд:

$$\begin{cases} p'_0(t) = \mu p_1(t) - \lambda p_0(t), \\ p'_1(t) = \lambda p_0(t) + 2\mu p_2(t) - (\lambda + \mu)p_1(t), \\ p'_2(t) = \lambda p_1(t) + 3\mu p_3(t) - (\lambda + 2\mu)p_2(t), \\ p'_3(t) = \lambda p_2(t) + 4\mu p_4(t) - (\lambda + 3\mu)p_3(t), \\ p'_4(t) = \lambda p_3(t) + 5\mu p_5(t) - (\lambda + 4\mu)p_4(t), \\ p'_5(t) = \lambda p_4(t) - 5\mu p_5(t). \end{cases}$$

3. Побудувати одночасно графіки всіх ймовірностей станів з аналітичної моделі для часу з проміжку $[0; 100]$.

4. Побудувати одночасно графіки аналітичної й імітаційної ймовірностей обслуговування для часу з проміжку $[0; 100]$.

5. Побудувати одночасно графіки аналітичної й імітаційної середньої кількості зайнятих каналів обслуговування для часу з проміжку $[0; 100]$.

6. Побудувати одночасно графіки аналітичного й імітаційного коефіцієнтів зайнятості системи масового обслуговування для часу з проміжку $[0; 100]$.

Лабораторна робота № 7. PLUS-процедури (двовимірний логнормальний закон розподілу)

Вхідний потік деталей на виробничу ділянку є пуассонівським з середнім значенням 3. Обробка деталей на виробничій ділянці здійснюється у два етапи. Під час кожного етапу може оброблятися не більше 5 деталей одночасно. Час обробки деталей є випадковою величиною зі зміщеним двовимірним логнормальним законом розподілу з середнім значенням 1,9 і середньоквадратичним відхиленням 1,1 для першого етапу і з середнім значенням 2,4 і середньоквадратичним відхиленням 0,6 для другого етапу, коефіцієнт кореляції дорівнює 0,7, зміщення для першого етапу становить 1, а для другого етапу – 1,5.

1. Створити модель на мові GPSS для моделювання роботи виробничої ділянки. Для моделювання зміщеного двовимірного логнормального закону розподілу потрібно створити універсальну процедуру `lognormal2`.

2. Визначити середні тривалості обробки 1000 деталей на першому і на другому етапах роботи.

3. Побудувати і проаналізувати графіки тривалостей обробки 20 деталей на першому і на другому етапах. Щоб можна було побачити взаємозв'язок, другий графік будувати з коефіцієнтом 0,5.

Вказівка. Для моделювання випадкової величини з двовимірним нормальним законом розподілу використовують формули: $x = \mu_x + \sigma_x z_1$, $y = \mu_y + \sigma_y r_{xy} z_1 + \sigma_y \sqrt{1 - r_{xy}^2} z_2$, де μ_x , μ_y – середні значення пов'язаних випадкових величин, σ_x , σ_y – їх середньоквадратичні відхилення, r_{xy} – кореляція, z_1 , z_2 – випадкові величини із стандартним нормальним законом розподілу.

Лабораторна робота № 8.

Робота з матрицями

Вхідний потік вимог задається неперервною функцією зі значеннями у вузлах $(0; 0)$, $(0, 2; 1)$, $(0, 5; 2)$, $(0, 7; 3)$, $(0, 8; 4)$, $(1; 7)$. Час обслуговування є рівномірно розподіленою на відрізок $[1; 5]$ випадковою величиною. Одночасно не може обслуговуватись більше однієї вимоги. Скласти матрицю, у перший рядок j -го стовпця якої буде заноситись час появи j -ї вимоги в моделі, а в другий рядок j -го стовпця – час перебування цієї вимоги в моделі. Здійснити моделювання обслуговування 100 вимог. Переглянути створену матрицю.

Найважливіші системні числові атрибути

$RN\alpha$ – генератор випадкових чисел α .

$AC1$ – абсолютний системний час.

$C1$ – відносний системний час (скидається командою **RESET**).

Абсолютний системний час командою **RESET** не перетворюється в нуль, але занулюється командою **CLEAR**.

$TG1$ – поточне значення лічильника завершень (задається командою **START**, а зменшується блоком **TERMINATE**).

Системні числові атрибути (СЧА) транзактів.

$P\alpha$ – значення параметра α активного транзакта.

PR – пріоритет активного транзакта.

$XN1$ – номер активного транзакта (натуральне число).

$M1$ – час перебування в моделі активного транзакта.

$MP\alpha$ – час проходження транзактом деякої частини моделі (різниця абсолютного модельного часу і значення параметра α транзакта: $AC1 - P\alpha$, це значення параметра α встановлюється блоком **MARK** α , що рівносильно **ASSIGN** $\alpha, AC1$).

СЧА блоків.

$N\alpha$ – загальна кількість входжень транзактів до блока з міткою α .

$W\alpha$ – поточна кількість транзактів у блоці з міткою α .

СЧА одноканальних пристроїв.

$F\alpha$ дорівнює 1, якщо одноканальний пристрій α зайнятий, інакше – 0.

$FI\alpha$ дорівнює 1, якщо обслуговування в одноканальному пристрої α було щойно перерване, інакше – 0.

$FV\alpha$ дорівнює 1, якщо одноканальний пристрій α доступний, інакше – 0.

$FR\alpha$ – коефіцієнт використання одноканального пристрою α (ціле число від 0 до 1000).

$FC\alpha$ – кількість транзактів, які займали одноканальний пристрій α .

$FT\alpha$ – середній час використання одноканального пристрою α одним транзактом.

СЧА багатоканальних пристроїв.

$S\alpha$ – кількість зайнятих каналів у багатоканальному пристрої α .

$R\alpha$ – кількість вільних каналів у багатоканальному пристрої α .

$SR\alpha$ – коефіцієнт використання багатоканального пристрою α (ціле число від 0 до 1000).

$SA\alpha$ – середня кількість зайнятих каналів у багатоканальному пристрої α .

$SM\alpha$ – максимальна кількість зайнятих каналів у багатоканальному пристрої α .

$SC\alpha$ – загальна кількість входжень транзактів до багатоканального пристрою α .

$ST\alpha$ – середній час використання одного каналу багатоканального пристрою α .

$SE\alpha$ дорівнює 1, якщо багатоканальний пристрій α порожній, інакше – 0.

$SF\alpha$ дорівнює 1, якщо багатоканальний пристрій α зайнятий повністю, інакше – 0.

$SV\alpha$ дорівнює 1, якщо багатоканальний пристрій α доступний, інакше – 0.

СЧА черг.

$Q\alpha$ – поточна довжина черги α .

$QA\alpha$ – середня довжина черги α .

$QM\alpha$ – максимальна довжина черги α .

$QС\alpha$ – загальна кількість входжень транзактів до черги α .

$QZ\alpha$ – кількість входжень транзактів з нульовим часом перебування їх у черзі α .

$QT\alpha$ – середній час перебування транзакта в черзі α .

$QX\alpha$ – середній час перебування транзакта (без нульових входжень) у черзі α .

СЧА таблиць.

$TV\alpha$ – середнє значення аргументу таблиці α .

$ТС\alpha$ – кількість введень інформації у таблицю α .

$TD\alpha$ – середнє квадратичне відхилення аргументу таблиці α .

СЧА комірок, змінних, функцій, ключів.

$X\alpha$ – вміст комірки пам'яті α .

$MX\alpha(a, b)$ – вміст комірки матриці α , що знаходиться на перетині рядка a і стовпця b .

$FN\alpha$ – результат обчислення функції α .

$V\alpha$ – результат обчислення цілочислової чи дійсної змінної α .

$BV\alpha$ – результат обчислення логічної змінної α .

$LS\alpha$ – стан логічного ключа α (1 – встановлений, 0 – не встановлений).

Приклад імітації виходу з ладу одного каналу

Для імітації виходу з ладу одноканального пристрою найкраще використовувати блок PREEMPT з пріоритетами, а імітувати несправність і її усунення (ремонт) за допомогою спеціального транзакта з високим пріоритетом.

Для імітації виходу з ладу одного чи кількох каналів багатоканального пристрою можна замість багатоканального пристрою використовувати кілька одноканальних пристроїв, входи в які треба моделювати блоками `PREEMPT` з пріоритетами, а імітувати несправність і її усунення спеціальним транзактом (транзактами) з високим пріоритетом.

Приклад. У магазині є два продавці. Час від часу до одного з них хтось телефонує і тоді продавець припиняє обслуговування покупця. Якщо вільний другий продавець, то покупець може звернутись до нього, або ж йому просто доведеться чекати доти, доки продавець наговориться по телефону і дасть здачу. Те саме – два верстати, один з яких виходить з ладу. Тоді деталь буде чекати завершення ремонту або звільнення другого верстату. Потік надходження покупців (деталей) є пуассонівським з середнім значенням 3. Час обслуговування – випадкова величина з логнормальним законом розподілу з середнім значенням 1 і середньоквадратичним відхиленням 1 для відповідного нормального розподілу. Вихід з ладу обладнання (продавця, верстату) моделюється логнормальним законом розподілу з середнім значенням 6 і середньоквадратичним відхиленням 2 відповідного нормального розподілу. Час ремонту – рівномірно розподілена на відрізьку [1; 3] випадкова величина.

```

tab table m1,1,1,100
generate (exponential(10,0,3))
assign 1,(lognormal(11,0,1,1))
mit1 queue cherga
test ne (f$prod1+f$prod2),2; чекати,
;доки не звільниться хоч один пристрій
depart cherga
gate nu prod1,mit3; якщо зайнятий пристрій 1, то йти
;до пристрою 2
mit2 preempt prod1,pr,mit1,1,re
advance p1
return prod1
transfer ,mit4

```

```
mit3 preempt prod2,pr,mit1,1,re
advance p1
return prod2
mit4 test e pr,0,mit5
tabulate tab
mit5 terminate 1
generate (lognormal(12,0,6,2)); генерування виходу
;з ладу пристрою
priority 1
assign 1,(uniform(13,1,3))
transfer 0.5,mit2,mit3
start 1000
```

Недоліком цієї моделі є те, що перерваний транзакт стає знову в хвіст черги. Щоб він ставав у голову черги, для нього можна використовувати проміжний пріоритет або використовувати спеціальну чергу з сортуванням транзактів по часу їх перебування в моделі. Якщо вилучити у блоках `preempt` три останніх операнди, то перерваний транзакт ставатиме у голову черги, але лише до того самого пристрою, навіть якщо другий пристрій вільний і немає ніяких черг. При імітації більшої кількості каналів все буде аналогічним.

Приклад імітації виходу з ладу багатоканального пристрою

Блок `DISPLACE A,B,C,D` призначений для відшукування в моделі потрібного транзакта і переміщення його до вказаного блока. Операнд *A* – номер транзакта, який потрібно перемістити. Транзакти нумеруються від 1, номер активного транзакта зберігається в системному числовому атрибуті (СЧА) `XN1`. Операнд *B* – мітка блока, до якого передається транзакт, вказаний операндом *A*. Операнд *C* – номер чи ім'я параметра цього транзакта, до якого записується час, що залишився до завершення його обслуговування у блоці `ADVANCE`, якщо транзакт перебував

у ньому. Блок `DISPLACE` спрацьовує при потраплянні до нього активного транзакта. Операнд D – мітка альтернативного блока для активного транзакта, якщо шуканий транзакт відсутній у моделі. Операнди C і D – необов'язкові. При відсутності операнда D активний транзакт переходить до наступного блока. Переміщення транзакта за межі приладу не звільняє прилад. Транзакт продовжує його займати.

Недоступність багатоканального пристрою моделює блок `SUNAVAIL A`, де A – ім'я багатоканального пристрою.

Зняття недоступності цього пристрою моделюється блоком `SAVAIL A`. Якщо у пристрої обслуговувались транзакти, то вони продовжуватимуть обслуговуватись, поки не вийде час. Інші транзакти зайти в недоступний пристрій не можуть. У блоці `GATE` умовний операнд `SV` використовуються для перевірки доступності пристрою, а `SNV` – його недоступності.

Блоки `FUNAVAIL A` і `FAVAIL A` призначені відповідно для моделювання недоступності і доступності одноканального пристрою. У блоці `GATE` умовний операнд `FV` використовуються для перевірки доступності пристрою, а `FNV` – його недоступності.

Імітацію виходу з ладу всього багатоканального пристрою здійснюють у ручному режимі за допомогою блоків `SUNAVAIL`, `SAVAIL`, `DISPLACE`.

Приклад. Вхідний потік вимог обслуговується у багатоканальному пристрої. Пристрій може вийти з ладу повністю. У випадку виходу з ладу багатоканального пристрою всі недообслужені вимоги ставляться у чергу для подальшого обслуговування у багатоканальному пристрої після його ремонту. Ємність пристрою – 10. Вхідний потік вимог є пуассонівським з середнім значенням 2. Час обслуговування – випадкова величина з логнормальним законом розподілу з середнім значенням 1 і середньоквадратичним відхиленням 1 відповідного нормального закону розподілу. Вихід з ладу багатоканального пристрою відбувається за логнормальним законом розподілу з середнім значенням 6 і середньоквадратичним відхиленням 2 відповідного нормального закону розподілу. Час ремонту бага-

токанального пристрою – рівномірно розподілена на проміжку [100; 200] випадкова величина.

```
bkp storage 10
tab table m1,10,10,100
generate (exponential(10,0,2))
assign 1,(lognormal(11,0,1,1))
mit1 queue cherga
enter bkp,1
depart cherga
assign zmc,10
mitc1 test e x*zmc,0,mitac1
savevalue p$zmc,xn1
transfer,mitvc1
mitac1 loop zmc,mitc1
mitvc1 advance p1
assign zmc,10
mitc2 test e x*zmc,xn1,mitac2
savevalue p$zmc,0
transfer,mitvc2
mitac2 loop zmc,mitc2
mitvc2 leave bkp,1
tabulate tab
terminate 1
generate (lognormal(12,200,6,2))
sunavail bkp
assign zmc,10
mitc3 test ne x*zmc,0,mitac3
displace x*zmc,mit2,1
savevalue p$zmc,0
mitac3 loop zmc,mitc3
advance 150,50
savail bkp
terminate 1
mit2 leave bkp,1
transfer,mit1
```

start 1000

Приклад створення спеціальної черги

При використанні одноканальних і багатоканальних пристроїв у моделі автоматично можуть виникати черги, які функціонують за принципом FIFO («першим прийшов – першим обслуговують») з врахуванням пріоритетів. Для створення черг з іншою дисципліною обслуговування потрібно використовувати списки користувача. Крім того, черги, які функціонують через списки користувача, у складних моделях працюватимуть швидше.

Список користувача є деяким буфером, до якого можуть тимчасово заноситись транзакти, причому вони вводяться до списку користувача і виводяться з нього відповідно до логіки моделі за допомогою спеціальних блоків.

Для введення транзактів до списку користувача слугує блок LINK A, B . Операнд A задає ім'я або номер списку користувача, до якого заноситься транзакт після входу в блок LINK. Операнд B визначає порядок перебування транзактів у списку користувача. Допустимі значення:

FIFO – транзакт передається в кінець списку;

LIFO – транзакт передається на початок списку;

$R\alpha$ – транзакти розташовуються в порядку зростання значення параметра транзактів α ;

СЧА, крім $R\alpha$, – транзакти розташовуються в порядку спадання значення СЧА.

Логічно використовувати в якості операнда B наступні СЧА:

PR – транзакти впорядковуються за спаданням пріоритету;

M1 – транзакти впорядковуються за спаданням часу перебування їх у моделі;

$R\alpha$ – транзакти впорядковуються за зростанням параметра транзактів α .

Кожен список користувача має такі СЧА:

$SN\alpha$ – поточна довжина списку користувача α ;

$SA\alpha$ – середня довжина списку α ;

$SM\alpha$ – максимальна довжина списку α ;

$SC\alpha$ – загальна кількість транзактів, які потрапили до списку α ;

$ST\alpha$ – середній час перебування транзактів у списку α .

Для виведення одного чи кількох транзактів зі списку користувача використовується блок `UNLINK X A,B,C,D,E,F`

Обов'язковими тут є лише операнди A і B . Операнд A задає ім'я або номер списку користувача. Операнд B – мітка блока, до якого переходять виведені зі списку користувача транзакти. Операнд C вказує кількість транзактів, які виводяться, або ключове слово `ALL` для виведення всіх транзактів, що є в списку. За замовчуванням, тобто коли не використовується операнд C , виводяться всі транзакти.

Якщо операнд D відсутній, то не використовуються також операнди X і E . Операнд D може бути: логічною змінною, ключовим словом `BACK`, ім'ям чи номером параметра транзакта.

Якщо операнд D є логічною змінною, то операнди X і E не використовуються. Логічна змінна підраховується відносно транзакта зі списку користувача. Якщо результат її обчислення не є нулем, тобто умова виведення виконується, то транзакт виводиться. Кількість транзактів, які виводяться, визначає операнд C . Але насправді може бути виведено менше транзактів, ніж вказано операндом C , за кількістю результатів обчислення логічної змінної, які не дорівнюють нулю. Крім того, транзактів у списку користувача може бути менше, ніж зазначено операндом C .

Якщо операндом D є ключове слово `BACK`, то операнди X і E також не використовуються, а транзакти виводяться з кінця списку в кількості, яка визначається операндом C .

В якості умовного операнда X використовуються ті самі значення, що й у випадку блока `TEST: E, NE, L, LE, G, GE`. Якщо операнд X пропущений, то вважається, що він рівний E . Якщо операнд D є ім'ям чи номером параметра транзакта і зада-

ний операнд E , то операнди D і E порівнюються відповідно до умовного операнда X . Якщо умова виконується, то транзакт виводиться зі списку користувача. Операнд D обчислюється відносно транзакта зі списку користувача.

Якщо операнд D задає параметр, а операнд E не використовується, то значення параметра транзакта зі списку користувача порівнюється зі значенням такого самого параметра транзакта, який ініціює виведення. Якщо вони рівні, то транзакт виводиться зі списку користувача.

Операндом F вказується мітка блока, куди переходить транзакт, який ініціює виведення, якщо зі списку користувача не було виведено жодного транзакта.

Приклад. В одноканальній системі масового обслуговування організувати таку дисципліну обслуговування, коли пріоритет віддається замовленням з найменшим часом обслуговування. Вхідний потік – пуассонівський з середнім часом 1. Час обслуговування – рівномірно розподілена випадкова величина на проміжку $[0,5; 2]$.

```
generate (exponential(5,0,1))
assign 1,(uniform(6,0.5,2))
gate nu pryl,wait
mit seize pryl
advance p1
release pryl
unlink cherga,mit,1
terminate 1
wait link cherga,p1
start 1000
```

При використанні списків користувача немає потреби вводити блоки для збирання статистики про чергу, бо всю необхідну статистику можна отримати у стандартному звіті з інформації про списки користувача. Зокрема, там наводиться максимальна довжина списку користувача, тобто черги, середній час перебування в черзі, поточна довжина черги.

Приклади інтегрування звичайних диференціальних рівнянь

Змінні користувача створюються командою опису EQU. Її формат: *ім'я* EQU *A*, де «*ім'я*» – це ім'я змінної, а операнд *A* – її значення. Значення змінних користувача можна змінювати за допомогою PLUS-процедур. За допомогою блоків змінні користувача змінювати не можна. Їх можна використовувати у блоках як константи.

У GPSS є можливість чисельно знаходити розв'язки початкових задач для звичайних автономних диференціальних рівнянь першого порядку та їх систем. Незалежна змінна у цих рівняннях має бути часом. Для інтегрування таких рівнянь використовуються команди опису INTEGRATE і EQU.

Формат команди INTEGRATE: *ім'я* INTEGRATE *A,B,C,D,E*. Тут «*ім'я*» – шукана функція $y(x)$ у рівнянні $y' = f(y)$, в якості імені беруть змінну користувача; операнд *A* – права частина диференціального рівняння, тобто значення похідної змінної користувача за часом; *B* – перше порогове значення шуканої функції; *C* – мітка блока, в який переходить згенерований транзакт у момент, коли шукана функція проходить через своє перше порогове значення у будь-якому напрямку; *D* – друге порогове значення шуканої функції; *E* – мітка блока, в який переходить згенерований транзакт у момент, коли шукана функція проходить через своє друге порогове значення у будь-якому напрямку. Операнди *B*, *C*, *D* і *E* – необов'язкові.

Початкова умова задається оголошенням початкового значення шуканої функції командою EQU. При цьому початкова умова задається в точці 0.

Для розв'язування системи диференціальних рівнянь потрібно використовувати кілька команд INTEGRATE і кілька команд EQU для відповідних початкових умов. Звичайне диференціальне рівняння n -го порядку, розв'язане відносно старшої похідної, зводиться до системи диференціальних рівнянь першого поряд-

ку:

$$y^{(n)} = F(x, y, y', \dots, y^{(n-1)}) \Rightarrow \begin{cases} y'_1 = y_2, \\ y'_2 = y_3, \\ \dots \\ y'_{n-1} = y_n, \\ y'_n = F(x, y_1, y_2, \dots, y_n). \end{cases}$$

Неавтономні рівняння і системи зводяться до автономних збільшенням кількості функцій і рівнянь, наприклад:

$$y' = f(x, y) \Rightarrow \begin{cases} y'_1 = 1, \\ y'_2 = f(y_1, y_2). \end{cases}$$

Розв'язок початкової задачі знаходиться числовим методом Рунге – Кутта – Фельберга п'ятого порядку. У кожен момент системного часу змінна користувача, відповідна шуканій функції, містить числовий розв'язок початкової задачі в цей момент часу. Знайдене значення розв'язку можна використовувати у блоках і в обчисленнях. Після завершення моделювання у стандартному звіті вказується значення, набуто змінною користувача в останній момент часу моделювання.

Приклад 1. У чан, що в початковий момент часу містить 10 л чистої води, неперервно надходить зі швидкістю 2 л за хвилину розчин, у кожному літрі якого міститься 0,3 кг солі. Цей розчин перемішується з водою і суміш витікає з чана з такою самою швидкістю. Скільки солі буде в чані через 10 хвилин?

Нехай $y(t)$ – кількість солі в чані в момент часу t , t вимірюється у хвилинах, $y(t)$ – у кг. Тоді

$$y(t + \Delta t) - y(t) = \left(2 \cdot 0,3 - 2 \cdot \frac{y(t + \alpha)}{10} \right) \Delta t$$

– приріст солі за час Δt . Поділивши на Δt , спрямувавши Δt до нуля і врахувавши, що $\alpha \rightarrow 0$ для $\Delta t \rightarrow 0$, отримаємо:

$$y' = 0,6 - 0,2y.$$

Крім того, маємо $y(0) = 0$. Для моделювання годиться модель:

```
y_ equ 0
y_ integrate (0.6-0.2#y_)
generate 10
terminate 1
start 1
```

Круглі дужки у другому рядку тут обов'язкові. У звіті вказується значення $y_$ на кінець моделювання, тобто для часу $t = 0$.

Приклад 2. Знайти час, коли в чані з попередньої задачі опиниться 29 кг солі.

```
y_ equ 0
y_ integrate (0.6-0.2#y_),2.9,mit
generate 1000
mit terminate 1
start 1
```

Час моделювання буде відповіддю на поставлену задачу.

Приклад 3. Знайти розв'язок системи $x' = -xy$, $y' = 2x - 6y$, $x = x(t)$, $y = y(t)$, $x(0) = 1$, $y(0) = 2$ в точці $t = 5$:

```
x_ equ 1
y_ equ 2
x_ integrate (-x#y_)
y_ integrate (2#x_-6#y_)
generate 5
terminate 1
start 1
```

Блок INTEGRATION A, B , де A – змінна користувача, описана командою INTEGRATE, а B є ключовим словом ON чи OFF, використовується для припинення інтегрування диференціального рівняння чи продовження його інтегрування. Значення OFF означає «припинити інтегрування», а значення ON – «відновити інтегрування». Припинення інтегрування приводить до того, що змінна користувача, описана командою INTEGRATE, буде містити розв'язок рівняння, відповідний часу входу будь-якого транзакта в блок INTEGRATION з цією змінною і словом OFF, і в майбутньому.

PLUS-процедури

У GPSS передбачено можливість створювати PLUS-процедури, які можна буде використовувати в арифметичних виразах у блоках як, наприклад, стандартні процедури – `Normal` (генерування нормально розподіленої випадкової величини) чи `sin` (синус числа). Оскільки ці процедури вертають значення, їх краще було б, по аналогії з багатьма мовами програмування, називати функціями.

Описується PLUS-процедура ключовим словом `PROCEDURE`, після якого ставиться ім'я створюваної процедури (послідовність букв, цифр, символів підкреслення), а далі у круглих дужках задаються формальні параметри. Якщо параметрів не потрібно, то круглі дужки ставити все одно треба.

Після заголовку процедури розміщується її тіло, яке обмежується ключовими словами `BEGIN` і `END`, причому після `END` ставиться крапка з комою (;). Замість великих букв скрізь можна вживати маленькі. Чітких обмежень на структуру рядка в процедурі немає, як і в багатьох мовах програмування. В рядку може бути і кілька операторів. Кожен оператор у тілі процедури закінчується крапкою з комою.

В тілі процедури можна використовувати змінні користувача, параметри процедури і системні числові атрибути. Крім того, можна створити локальні змінні процедури командою `TEMPORARY`, після якої всі ці змінні перераховуються через кому. В кінці їх, як після кожного оператора, ставиться крапка з комою. Команда `TEMPORARY` розміщується в тілі процедури (після ключового слова `BEGIN`). Локальна змінна, як і вузьке ім'я `СЧА` та змінна користувача, – це послідовність букв, цифр і символів підкреслення, перший символ імені – буква. Ім'я локальної змінної має відрізнитись від ключових слів і класів `СЧА`.

Командою `TEMPORARY MATRIX` можна оголошувати також локальні матриці. Тут все робиться аналогічно до локальних змінних, але у квадратних дужках після імені кожної матриці через кому перелічується кількість елементів матриці за ко-

жною розмірністю.

Оператор присвоювання записується у формі:

змінна=*вираз*;

Присвоювати значення можна лише локальним змінним процедури, змінним користувача і елементам матриць. Присвоювати значення СЧА, крім матриць, зокрема, коміркам пам'яті чи параметрам транзакта, не можна. СЧА можуть використовуватись тільки у правій частині оператора присвоювання. Вирази будуються за правилами побудови арифметичних виразів у GPSS, зокрема # позначає множення. У круглі дужки брати вирази тут необов'язково. У виразах можна викликати інші процедури, зокрема стандартні. Присвоювати значення можна також елементам матриці, але при цьому в лівій частині оператора присвоювання записується лише вузьке ім'я матриці (без класу СЧА МХ), а елементи матриці записуються у квадратних дужках через кому, наприклад, $abcd[2,3,4]=3+abcd[1,1,2]$; . У правій частині оператора присвоювання матриці задаються аналогічно. Пригадаємо, що в блоках для індексів матриць використовують круглі дужки і ставлять клас СЧА МХ.

Оператор розгалуження будується так:

IF (*логічний вираз*) THEN *оператор*;

або

IF (*логічний вираз*) THEN *оператор* ELSE *оператор*;

Логічний вираз будується за правилами побудови логічних виразів у GPSS і береться у круглі дужки. Замість одного оператора після ключового слова THEN чи ELSE можна записувати послідовність операторів, відокремлених крапками з комами, яка записується між ключовими словами BEGIN і END.

Оператор циклу:

WHILE (*логічний вираз*) DO *оператор*;

Тут також замість одного оператора може бути їх послідовність між ключовими словами BEGIN і END.

Оператор переходу на мітку всередині процедури:

GOTO *мітка*;

Мітка – це послідовність букв і цифр, що починається з

букви. Мітка відокремлюється від оператора, на який потрібно перейти, символом двокрапки. Наприклад,

```
...
GOTO sobaka;
...
sobaka:t=12;
```

```
...
Оператор повернення значення:
RETURN вираз;
```

Обчислене значення виразу вертається як результат роботи процедури, оператори після цього ігноруються, відбувається вихід з процедури. Можна писати просто RETURN; У останньому випадку результатом буде 0.

Виклик PLUS-процедури використовують для обчислення виразу у потрібному блоці. Якщо важливим є не саме значення, яке обчислює процедура, а якісь інші «побічні» результати її роботи, то результат роботи процедури можна присвоїти якомусь параметру транзакта чи комірці пам'яті, які потім не використовуватимуть. Є спеціальний блок PLUS A який викликає процедуру, що ставиться у круглих дужках замість операнда A, але не присвоює нікуди результат її роботи.

Приклади використання PLUS-процедур

Приклад 1. Знайти розв'язок початкової задачі $y' = 0,6 - 0,2y$, $y(1) = 0,5$ в точці $t = 10$.

```
y_ equ 0
y_ integrate (0.6-0.2#y_)
procedure newint(valuei)
begin
y_=valuei;
end;
generate 1,,1
plus(newint(0.5))
```

```
advance 9
terminate 1
start 1
```

У момент часу, рівний одиниці, генерується один транзакт, який запускає PLUS-процедуру зміни значення змінної користувача у_. Процес інтегрування продовжується до часу $t = 10$, коли знищується транзакт і припиняється моделювання.

Приклад 2. Нехай, наприклад, є нестационарний пуассонівський потік з інтенсивністю

$$\lambda(t) = \begin{cases} 5, & 0 \leq t \leq 10, \\ 1, & 10 < t \leq 20, \\ 6, & 20 < t \leq 30, \\ 2, & 30 < t \leq 40, \\ 5, & 40 < t \leq 50, \end{cases} \quad \lambda(t + 50) = \lambda(t).$$

Модель для його імітування на основі методу проріджування може бути такою:

```
ttt equ 0
proc function ttt,d5
10,5/20,1/30,6/40,2/50,5
procedure nestac_poisson2() begin
temporary t,rr2,lambda,t50;
t=ac1;
rr2=1;
lambda=0;
while (rr2>lambda/6) do
begin
t=t+exponential(5,0,1/6);
t50=int(t/50);
ttt=t-t50#50;
lambda=fn$proc;
rr2=uniform(6,0,1);
end;
return (t-ac1);
end;
```

```

tab table ac1,10,10,3000
generate (nestac_poisson2())
tabulate tab
terminate 1
start 1000

```

Матриці

Можна створювати матриці комірок пам'яті. Для опису матриці використовується команда **MATRIX**. Формат запису команди:

ім'я **MATRIX** *A,B,C*

Операнд *A* не використовується (залишений для сумісності зі старими версіями програми). Операнд *B* – число рядків матриці, *C* – число стовпців матриці.

Початкові значення елементам матриці можуть бути присвоєні командою **INITIAL** з використанням індексної форми запису чи без неї. Наприклад:

ABCD MATRIX ,2,2

INITIAL **MX\$ABCD**(1,2),5 – елементу матриці **ABCD** з індексами (1,2) присвоюється число 5.

INITIAL **ABCD**,10 – всі елементи матриці будуть містити 10.

MX – клас СЧА для матриць.

Мхім'я(*a,b*) – форма звертання до елемента матриці. Наприклад, **MX\$ABCD**(2,1).

Блок **MSAVEVALUE** використовується для запису значень у матриці, а також для збільшення чи зменшення значень елементів матриці. Його формат:

MSAVEVALUE *A,B,C,D*

або

MSAVEVALUE *A+,B,C,D*

або

MSAVEVALUE *A-,B,C,D*

Тут A – ім'я матриці, B – номер рядка матриці, C – номер стовпця матриці, D – значення, за допомогою якого відбувається модифікація початкового значення.

Матриці можуть мати розмірність більшу двох (три, чотири, п'ять чи шість). Для цього в команді опису використовують додаткові операнди:

ім'я MATRIX A,B,C,D,E,F,G

Присвоїти значення елементу матриці з розмірністю, більшою двох, можна за допомогою спеціальних процедур користувача. Блок `MSAVEVALUE` дозволяє здійснювати присвоєння елементам матриці з одиницями в індексах для всіх розмірностей, вище другої.

Налагодження моделі

Для налагодження моделі можна використовувати кілька спеціальних команд керування і вікна, які дозволяють відстежувати процес моделювання. Спочатку розглянемо команди керування. Їх дають за допомогою меню **Command**. Для їх виконання призначені також певні комбінації клавіш.

HALT припиняє процес моделювання. Клавіша виклику – **F4**.

CONTINUE продовжує процес моделювання, якщо лічильник кількості завершень не дорівнює нулю. Може використовуватись після команди **HALT**, а також при призупиненні моделювання командою **STOP**. Клавіша виклику – **F2**.

START A, де A – лічильник кількості завершень, – запуск процесу моделювання.

STEP A – виконується процес моделювання заданої кількості входжень транзактів до блоків. Операнд A – задана кількість входжень транзактів до блоків.

STEP 1 використовується для покрокового виконання процесу моделювання. Клавіша виклику – **F5**.

STOP A,B,C використовується для встановлення або знімання умови переривання процесу моделювання. Команду можна записувати в тексті моделі, як і команду **START**. Команди **STOP**

немає в меню **Command**, але в меню **Command** є пункт **Custom**, за допомогою якого можна давати будь-яку команду керування. Операнди: *A* – номер транзакта (нумеруються від одиниці); *B* – номер або мітка блока, *C* – визначає стан команди, може бути **ON** чи **OFF**, за замовчуванням **ON**. Всі операнди є необов'язковими, але принаймні один має бути.

Приклад. **STOP 13,7** – задається умова, яка приведе до зупинки процесу моделювання в той момент, коли транзакт номер 13 намагатиметься увійти до блока номер 7. Після цього дають команду **START** з достатнім значенням лічильника завершення. Якщо була виконана умова зупинки, то процес моделювання призупиняється й у рядок стану і вікно **JOURNAL** передається системний час, номер активного транзакта, а також блок, в якому він перебуває.

Для продовження моделювання використовують команду **CONTINUE**. Вона не знімає умови зупинки. Якщо ця умова знову виконається, то процес моделювання знову зупиниться. Для зняття всіх умов зупинки використовують команду **STOP „OFF**.

SHOW A – в рядку стану записується результат обчислення виразу *A*. Процес моделювання має бути призупиненим. Може використовуватись для перегляду змінних, комірок пам'яті, логічних ключів, системних числових атрибутів на будь-якому кроці моделювання. Пов'язана з командами **STEP**, **STOP** і **HALT**.

Приклади:

SHOW XN1 – показується номер активного транзакта.

SHOW N\$Mit1/N\$Mit2 – виводиться результат ділення кількості транзактів, що увійшли до блока з міткою *Mit1*, на кількість транзактів, які увійшли до блока з міткою *Mit2*.

Для спостереження за процесом моделювання на етапі тестування і верифікації використовують десять динамічних вікон. Доступ до них відкриває команда з меню **Window►Simulation Window**. Існують наступні вікна:

- **Blocks** (блоки);
- **Expression** (вираз);
- **Facilities** (одноканальні пристрої);

- **Logicswitches** (логічні ключі);
- **Matrix** (матриця);
- **Plot** (графік);
- **Queues** (черги);
- **Savevalues** (комірки пам'яті);
- **Storages** (багатоканальні пристрої);
- **Table** (таблиця).

Ці вікна можна переглядати як під час зупинки процесу моделювання чи після його завершення, так і під час процесу моделювання. В останньому випадку моделювання суттєво сповільнюється, але інформація у вікнах оперативного оновлюється відповідно до процесу моделювання.

Вікно **Blocks** є одним з найважливіших. Воно дає можливість візуально відстежувати пересування транзактів між блоками.

Вікно **Expression** призначене для спостереження за змінами будь-якої кількості виразів. Команда **SHOW** обчислює один вираз у конкретний момент часу, а вікно **Expression** дозволяє спостерігати зміни кількох виразів під час моделювання.

Вікно **Plot** дає змогу одночасно спостерігати за кількома графіками зміни заданих виразів.

Вікно **Table** є динамічною гістограмою, корисною для спостереження за збиранням даних.

У вікнах **Blocks**, **Facilities**, **Logicswitches**, **Queues**, **Savevalues**, **Storages** є кнопки **Halt**, **Continue** і **Step 1**, які можна використовувати для покрокового моделювання і перегляду вікон одночасно з цим.

Крім перегляду динамічних вікон можна робити кадри стану (**Window►Simulation Snapshot**) для:

- **CEC** (списку поточних подій);
- **FEC** (списку майбутніх подій);
- **Transaction** (окремого транзакта);
- **Numeric Groups** (числових груп);
- **Transactions Groups** (груп транзактів);
- **Userchains** (списків користувача).

Список рекомендованої літератури

1. Боев В. Д. Моделирование систем. Инструментальные средства GPSS World / В. Д. Боев. – СПб : БХВ-Петербург, 2004. – 368 с.
2. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування. Практикум / Ю. В. Жерновий. – Львів : Вид. центр ЛНУ ім. І. Франка, 2007. – 307 с.
3. Кудрявцев Е. М. GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – М. : ДМК Пресс, 2004. – 320 с.
4. Махней О. В. Лабораторний практикум з імітаційного моделювання у GPSS / О. В. Махней. – Івано-Франківськ : ВДВ ЦІТ, 2010. – 36 с.
5. Томашевский В. Имитационное моделирование в среде GPSS / В. Томашевский., Е. Жданова. – М. : Бестселлер, 2003. – 416 с.
6. Шрайбер Т. Дж. Моделирование на GPSS / Т. Дж. Шрайбер. – М. : Машиностроение, 1980. – 593 с.