

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка

Ю. В. Жерновий

**ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ
СИСТЕМ
МАСОВОГО ОБСЛУГОВУВАННЯ**

Практикум

Львів
Видавничий центр ЛНУ імені Івана Франка
2007

УДК 519.217(042.4)

ББК В 171я73-2

Ж 59

Рецензенти:

д-р фіз.-мат. наук, проф. *Б. І. Копитко* (Львівський національний університет імені Івана Франка);

канд. фіз.-мат. наук *Р. О. Гринів* (Інститут прикладних проблем механіки та математики імені Я. С. Підстригача НАН України)

*Рекомендовано до друку Вченою радою
механіко-математичного факультету
Протокол №8 від 22.03.2006*

Жерновий Ю. В.

Ж 59 Імітаційне моделювання систем масового обслуговування: Практикум. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 307 с.

Викладено основи імітаційного моделювання систем масового обслуговування з використанням мови GPSS та її реалізації GPSS World. На численних прикладах розкрито методи побудови імітаційних моделей за допомогою інструментальних засобів GPSS World. Значну увагу зосереджено на прикладах порівняння результатів імітаційного та аналітичного моделювання систем масового обслуговування і використання можливостей GPSS World для числового інтегрування систем диференціальних рівнянь Колмогорова для ймовірностей станів. Розглянуто моделі, які слугують для дослідження наближення сумарних і розріджених потоків подій до найпростішого потоку.

Для студентів математичних спеціальностей університетів та фахівців, які моделюють функціонування різноманітних систем в умовах невизначеності.

УДК 519.217(042.4)

ББК В 171я73-2

© Жерновий Ю. В., 2007

© Львівський національний університет імені Івана Франка, 2007

ВСТУП

Моделювання – найпотужніший універсальний метод дослідження та оцінювання ефективності різноманітних систем, поведінка яких залежить від дії випадкових чинників.

Аналітичне моделювання полягає у побудові та дослідженні математичних моделей. У його основу покладено ідентичність форми рівнянь та однозначність співвідношень між змінними в рівняннях, які описують оригінал та модель.

Недоліками більшості аналітичних моделей, побудованих на основі понять теорії масового обслуговування, є використання в них значних спрощень: зображення потоку замовлень як пуассонівського або найпростішого, припущення про показниковий розподіл часу обслуговування, неможливість обслуговування замовлень одночасно кількома каналами обслуговування тощо. Такі спрощення, а іноді штучне пристосування аналітичних моделей з метою використання добре розробленого математичного апарату для дослідження реальних систем можуть ставити під сумнів результати аналітичного моделювання. Недоліком складних моделей є громіздкість обчислень. Зокрема, аналітичний розв'язок системи диференціальних рівнянь Колмогорова для ймовірностей станів системи масового обслуговування можна знайти лише у випадку, коли кількість каналів обслуговування не перевищує двох. Складною для розв'язування у таких випадках є й відповідна система алгебричних рівнянь для ймовірностей станів граничного стаціонарного режиму. Отже, аналітичні методи мають самостійне значення лише для дослідження функціонування систем масового обслуговування у першому наближенні і в окремих, специфічних задачах.

На відміну від аналітичного імітаційне моделювання знімає більшість обмежень, пов'язаних з можливістю відображення в моделях реального процесу функціонування системи, яку досліджують, динамічної взаємної обумовленості поточних і наступних подій, комплексного взаємозв'язку між параметрами і показниками ефективності системи тощо. Хоч імітаційні моделі в деяких випадках не такі лаконічні, як аналітичні, проте вони можуть бути як завгодно близькими до системи, яку моделюють, і простими у використанні. Це дає змогу застосовувати імітаційне моделювання як універсальний підхід для прийняття рішень в умовах невизначеності, враховуючи в моделях навіть ті чинники, які важко формалізувати, а також використовувати головні принципи системного підходу для розв'язування практичних задач.

Імітаційні моделі описують об'єкт дослідження деякою мовою, імітуючи елементарні явища, з яких складається функціонування системи, зі збереженням їхньої логічної структури, послідовності протікання у часі, особливостей і складу інформації про стан процесу. Зазначимо про наявність аналогії між дослідженням процесів методом імітаційного моделювання та їхнім експериментальним дослідженням.

Описи компонентів реальної системи в імітаційній моделі мають певний логіко-математичний характер і є сукупністю алгоритмів, які імітують функціонування цієї системи. Програма моделі, побудована на основі цих алгоритмів, дає змогу звести імітаційне моделювання до проведення експериментів на ЕОМ шляхом їхнього „прогону” на деякій множині вхідних даних, які імітують первинні події, що відбуваються в системі. Інформація, яка фіксується у процесі дослідження імітаційної моделі, дає змогу визначити потрібні показники, що характеризують ефективність системи, яку досліджують.

Завдяки застосуванню універсальних мов програмування для реалізації імітаційних моделей дослідник досягає гнучкості під час створення, відлагодження та випробування моделі. Однак мови моделювання, орієнтовані на певну предметну область, є мовами вищого рівня, тому дають підстави з меншими витратами створювати програми моделей для дослідження складних систем.

Одною з перших спеціалізованих мов моделювання, що полегшила процес написання імітаційних програм, була мова GPSS (General Purpose Simulation System – система моделювання загального призначення), яку створив у вигляді кінцевого продукту Джефрі Гордон у фірмі IBM 1962 р. Вона у свій час входила до першої десятки кращих мов програмування, випереджаючи транслятор з мови АЛГОЛ, і була реалізована практично на всіх типах ЕОМ. Сьогодні є транслятори для операційних систем DOS – GPSS/PC, для OS/2 і DOS – GPSS/H і для Windows – GPSS World.

Систему GPSS World розробила для OS Windows компанія Minuteman Software (США) у 2000 р. Це – потужне універсальне середовище моделювання як дискретних, так і неперервних процесів, призначене для професійного моделювання найрізноманітніших процесів і систем.

Порівняно з GPSS/PC у GPSS World з’явилися додаткові можливості. Головними з них є:

- для всіх класів об’єктів і змінних реалізовано динамічні графічні вікна, в яких у реальному часі подано проміжну і вихідну статистику;
- гнучку процедурну мову PLUS можна використати для побудови моделей і в процедурах здійснення експерименту;
- введено засоби підтримки факторного аналізу, традиційного дисперсійного (ANOVA) і регресійного аналізу, оптимізацію на основі методології оптимального планування експерименту;
- стали доступними елементи неперервного моделювання;
- вирішено проблеми з цілочисельним модельним часом.

За допомогою цієї системи можна ефективно моделювати як виробничі, так і невиробничі процеси: функціонування закладів торгівлі, портів, вуличний рух, роботу редакцій, установ, комп’ютерних мереж і мережі Internet, різноманітних систем масового обслуговування тощо. Система має великий

набір команд для керування процесом моделювання, які можна використовувати як в інтерактивному режимі, так і вводити до складу моделі. Забезпечено можливість здійснення експериментів, згенерованих системою, у тім числі оптимізаційних. У системі GPSS World реалізовано процедуру візуалізації процесу функціонування моделі з використанням методів мультиплікації.

Система GPSS World функціонує в режимі транслятора, в той час, як GPSS/PC – у режимі інтерпретатора, тобто пострічкового виконання програми. Ще одна суттєва відмінність GPSS World від GPSS/PC полягає в тому, що у новій програмі немає необхідності нумерувати рядки програми.

Система містить велику кількість різних типів вікон, що спрощує перегляд та аналіз об'єктів моделі. У GPSS World є бібліотека розподілів імовірностей, які можна широко використовувати в процесі моделювання. Крім того, є бібліотека процедур, яка забезпечує маніпуляції з рядковими даними, дає змогу здійснювати розрахунки і використовувати розподіли ймовірностей.

Використання системи моделювання GPSS World не лише значно прискорює процес моделювання та дослідження найрізноманітніших систем масового обслуговування і неперервних процесів, а й дає змогу здійснювати оптимізаційні експерименти.

Сьогодні майже немає навчального матеріалу українською мовою для вивчення мов моделювання взагалі і GPSS World, зокрема.

Мета навчального посібника – ознайомлення читача з основами імітаційного моделювання систем масового обслуговування з використанням мови GPSS та її реалізації GPSS World.

У першому розділі розглянуто класифікацію систем масового обслуговування та основи організації імітаційного моделювання. Основи побудови та загальну характеристику об'єктів мови GPSS викладено у другому розділі. Третій розділ присвячений вивченню принципів функціонування основних блоків GPSS World, які використовують для побудови моделей систем масового обслуговування.

У четвертому розділі описано процес створення об'єктів GPSS World та експлуатації моделей.

У п'ятому та шостому розділах на численних прикладах, які розробив автор, розкрито методи побудови моделей процесів, які формалізуються у вигляді схем масового обслуговування. Сьомий розділ присвячено побудові моделей для дослідження сумарних і розріджених потоків подій з погляду їхнього наближення до найпростішого потоку.

1. ОСНОВИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

1.1. Системи масового обслуговування та їхні характеристики

Із системами масового обслуговування (СМО) ми зустрічаємось повсякчас. Кожному з нас доводилось чекати обслуговування в черзі (у магазині, на автозаправці, в бібліотеці, кав'ярні тощо). Аналогічні ситуації виникають, коли треба скористатися телефонним зв'язком або виконати свою програму на комп'ютері. Будь-яке виробництво теж можна уявити як послідовність систем обслуговування. До типових систем обслуговування належать також ремонтні і медичні служби, транспортні системи, аеропорти, вокзали тощо.

Особливого значення набули такі системи у процесах інформатики. Це передусім комп'ютерні системи, мережі передавання інформації, операційні системи, бази і банки даних. Системи обслуговування відіграють значну роль у повсякденному житті. Досвід моделювання різних типів дискретних систем свідчить про те, що приблизно 80% цих моделей ґрунтуються на СМО [7].

Систему масового обслуговування загалом можна уявити як сукупність послідовно пов'язаних між собою вхідних потоків вимог на обслуговування (потоків замовлень), черг, каналів обслуговування і потоків обслужених замовлень. Будь-який пристрій, який безпосередньо обслуговує замовлення, називають *каналом обслуговування*.

Системи масового обслуговування за наявності тої чи іншої ознаки можна класифікувати так [6]:

1. *За характером надходження замовлень у систему*: на системи з регулярним і випадковим потоками замовлень. Якщо кількість замовлень, які надходять у систему за одиницю часу (інтенсивність потоку), стала або є заданою функцією часу, то маємо систему з регулярним потоком замовлень, в іншому разі – з випадковим. Випадковий потік замовлень може бути *стаціонарним* або *нестационарним*. Якщо параметри потоку замовлень не залежать від розташування інтервалу часу, який розглядають, на осі часу, то маємо стаціонарний потік замовлень, в протилежному випадку – нестационарний. Наприклад, якщо кількість покупців, які приходять до магазину, не залежить від часу доби, то потік замовлень (покупців) – стаціонарний.
2. *За кількістю замовлень, які надходять за одиницю часу*: на системи з ординарним і неординарним потоками замовлень. Якщо ймовірність надходження двох або більше замовлень в один момент часу дорівнює нулеві або настільки мала, що нею можна знехтувати, то маємо систему з *ординарним потоком замовлень*. Наприклад, потік літаків, які прибувають на злітну смугу аеродрому (ЗС), можна вважати ординарним, оскільки ймовірність надходження двох і більше літаків до

- каналу обслуговування (ЗС) в один і той самий момент часу дуже мала.
3. *За зв'язком між замовленнями*: на системи без післядії від замовлень, які надійшли, і з післядією. Якщо ймовірність надходження замовлень у систему в деякий момент часу не залежить від того, скільки вимог уже надійшло до системи, тобто не залежить від передісторії процесу, який вивчають, то ми маємо задачу *без післядії*, у протилежному випадку – з *післядією*. Прикладом задачі з післядією може слугувати потік студентів на складання заліку викладачеві.
 4. *За характером поведінки замовлень у системі*: з відмовами, з обмеженим очікуванням і з очікуванням без обмеження:
 - якщо нове замовлення, яке прибуло на обслуговування, застає усі канали обслуговування уже зайнятими і покидає систему, то маємо *систему з відмовами*. Замовлення може покинути систему і тоді, коли черга досягла певних розмірів. Якщо ракета супротивника з'являється в час, коли всі протиракетні пристрої обслуговують інші ракети, то вона без проблем залишає зону обслуговування;
 - якщо нове замовлення, яке прибуло на обслуговування, застає усі канали обслуговування зайнятими і стає у чергу, але перебуває у ній обмежений час і, не дочекавшись обслуговування, покидає систему, то маємо *систему з обмеженим очікуванням*. Прикладом такого „нетерплячого” замовлення може бути самоскид із цементним розчином. Якщо час очікування великий, то щоб запобігти затвердненню розчину, він може бути розвантажений в іншому місці;
 - якщо нове замовлення, яке прибуло на обслуговування, заставши усі канали обслуговування зайнятими, змушене очікувати своєї черги до того часу, поки не буде обслужене, то маємо *систему з очікуванням без обмеження*. Приклад: літак, який перебуває на аеродромі до того часу, поки не звільниться злітна смуга.
 5. *За способом вибору замовлень на обслуговування*: з пріоритетом, за часом надходження, випадково, останнього обслуговують першим. Іноді в такому випадку кажуть про дисципліну обслуговування:
 - якщо система масового обслуговування охоплює кілька категорій замовлень і з певних міркувань необхідно дотримуватись різного підходу до їхнього відбору, то маємо *систему з пріоритетом*. Зокрема, під час надходження виробів на будмайданчик, перш за все монтують ті, які необхідні у цей момент;
 - якщо канал, який звільнився, обслуговує замовлення, яке раніше за інших надійшло до системи, то маємо систему з обслуговуванням замовлень *за часом надходження*. Це найпоширеніший клас систем. Наприклад, покупець, який підійшов до продавця першим, обслуговують раніше за інших. Цей спосіб вибору замовлень на обслуговування застосовують там, де внаслідок технічних, технологічних або організаційних умов замовлення не можуть випереджати одне одного;

- якщо замовлення з черги надходять до каналу обслуговування у випадковому порядку, то маємо систему з *випадковим вибором замовлень на обслуговування*. Приклад: вибір слюсарем-сантехніком одного з декількох замовлень на усунення несправностей, які надійшли від мешканців. Вибір тут, зазвичай, визначають місцезнаходженням самого слюсаря: він надасть перевагу замовленню мешканця, який перебуває від нього найближче, якщо інші чинники не визначають вибору;
- *останнього обслуговують першим*. Цей спосіб вибору вимог на обслуговування використовують у тих випадках, коли зручніше й економніше брати на обслуговування замовлення, яке найпізніше надійшло до системи. Зокрема, якщо будівельні вироби складені один на одному, то зручніше спочатку брати виріб, який надійшов останнім.
6. *За характером обслуговування замовлень*: на системи з детермінованим і випадковим часом обслуговування. Якщо інтервал часу між моментами надходження замовлення до каналу обслуговування і моментом виходу замовлення з цього каналу є сталим, то йдеться про систему з *детермінованим часом обслуговування*, в іншому разі – з *випадковим*.
7. *За кількістю каналів обслуговування*: на одноканальні і багатоканальні системи. Наприклад, для зведення будинку можна використати один будівельний кран (*один канал обслуговування*) або декілька (*багато каналів*) для обслуговування виробів, які прибувають на будову.
8. *За кількістю етапів обслуговування*: на однофазні і багатофазні системи. Якщо канали обслуговування розташовані послідовно, і вони неоднорідні, оскільки виконують різні операції обслуговування, то йдеться про *багатофазну* систему масового обслуговування. Прикладом такої системи може бути обслуговування автомобілів на станції технічного обслуговування (миття, діагностування тощо).
9. *За однорідністю замовлень*, які надходять на обслуговування: на системи з однорідними і неоднорідними потоками замовлень. Наприклад, якщо для розвантаження прибувають фургони однакової вантажомісткості, то такі замовлення називають *однорідними*, якщо різної – то *неоднорідними*.
10. *За обмеженістю потоку замовлень*: на замкнені і розімкнені системи. Якщо потік замовлень обмежений і замовлення, які покинули систему, через деякий час до неї повертаються, то маємо *замкнену* систему, в протилежному випадку – *розімкнену*. Прикладом замкненої ситеми може слугувати бригада робітників, які налагоджують станки в ткацькому цеху.

З метою скорочення запису для позначення будь-якої однофазної СМО використовують систему кодування A/B/C/D/E, де на місці латинської літери ставлять відповідні характеристики системи:

А – закон розподілу інтервалів між надходженнями замовлень. Найчастіше використовують такі закони розподілу: показниковий (М), ерлангівський (Е), гіперекспоненціальний (Н), гамма-розподіл (Г), детермінований (D). Для позначення довільного характеру розподілу використовують символ G;

В – закон розподілу часу обслуговування в каналах СМО. Тут використовують такі самі позначення, як і для розподілу інтервалів між надходженнями замовлень;

С – кількість каналів обслуговування. Тут використовують такі позначення: для одноканальних систем записують 1, для багатоканальних l (кількість каналів);

D – кількість місць у черзі. Якщо кількість місць у черзі необмежена, то це позначення можна не використовувати. Для скінченної кількості місць у черзі в загальному випадку приймають позначення r або n (кількість місць);

Е – дисципліна обслуговування. Найчастіше використовують такі варіанти системи обслуговування: FIFO (першим прийшов – першим вийшов), LIFO (останнім прийшов – першим вийшов), RANDOM (випадковий порядок обслуговування). За дисципліни FIFO це позначення можна не використовувати.

Приклади позначень:

M/M/1 – СМО з одним каналом обслуговування, нескінченною чергою, показниковими законами розподілу інтервалів часу між надходженнями замовлень і часу обслуговування та дисципліною обслуговування FIFO;

E/H//r/LIFO – СМО з кількома каналами обслуговування, скінченною чергою, ерлангівським законом розподілу інтервалів часу між надходженнями замовлень, гіперекспоненціальним розподілом часу обслуговування та дисципліною обслуговування LIFO;

G/G/l – СМО з кількома каналами обслуговування, нескінченною чергою, довільними законами розподілу інтервалів часу між надходженнями замовлень і часу обслуговування та дисципліною обслуговування FIFO.

Вивчення або задання потоку замовлень, механізму (кількості каналів, тривалості обслуговування тощо) та дисципліни обслуговування дає підстави для побудови моделі системи.

2. ОСНОВИ ПОБУДОВИ МОВИ GPSS

2.1. Загальні відомості про мову GPSS. Об'єкти системи GPSS World

Система GPSS World призначена для імітаційного моделювання систем з дискретними та неперервними процесами. У ній мова моделювання GPSS покращена вбудованою мовою програмування низького рівня PLUS. Мова GPSS ґрунтується на припущенні, що модель складної системи можна зобразити за допомогою сукупності елементів і логічних правил їхньої взаємодії у процесі функціонування системи. Також припускається, що в процесі моделювання системи можна виокремити невеликий набір абстрактних елементів, які називають *об'єктами*. Набір логічних правил обмежений і його можна описати невеликою кількістю стандартних операцій. Комплекс програм, які описують функціонування об'єктів і виконують логічні операції, є основою для створення програмної моделі системи цього класу.

У складі системи GPSS World є також спеціальна *програма-планувальник*, яка виконує такі функції:

- забезпечення просування за заданими розробником маршрутами динамічних об'єктів, які називають *транзактами*;
- планування подій, які відбуваються в моделі, шляхом реєстрації часу появи кожної події і виконання їх у наростаючій часовій послідовності;
- реєстрація статистичної інформації про функціонування моделі;
- просування модельного часу у процесі моделювання системи.

Щоб забезпечити правильну послідовність обробки подій у часі, існує системний годинник, який зберігає значення абсолютного модельного часу. Всі проміжки часу описують додатними числами.

Вихідну програму мовою GPSS World, як і програму будь-якою мовою програмування, складають з послідовності *операторів*.

Об'єкти в системі, яку моделюють, мають різноманітні призначення і залежать від характеристик моделі, а також у деяких випадках, від розробника моделі. Обов'язкова наявність блоків і транзактів, інакше модель працювати не буде.

Об'єкти класифікують на *сім категорій* і *15 типів*, які наведені у табл. 2.1. Розглянемо призначення об'єктів GPSS.

Таблиця 2.1

Об'єкти GPSS

Категорії	Типи об'єктів
Динамічна	Транзакти
Операційна	Блоки
Апаратна	Одноканальні пристрої,

	багатоканальні пристрої, логічні ключі
Обчислювальна	Змінні, функції, генератори випадкових чисел
Статистична	Черги, таблиці
Запам'ятовуюча	Комірки, матриці комірок
Групуюча	Числові групи, групи транзактів, списки

Динамічними об'єктами є транзакти, які створюються у певних точках моделі, просуваються планувальником через блоки, а потім знищуються. Транзакти є аналогами замовлень у реальній СМО. З кожним транзактом може бути пов'язана певна кількість параметрів, які містять необхідну інформацію про нього. Параметри нумерують або їм дають імена. Нумери параметрів та імена використовують для посилань на значення, присвоєні параметрам. Транзакти можуть мати різні пріоритети. Пріоритет визначає перевагу, яку отримує транзакт, коли він та інші транзакти претендують на один і той самий ресурс.

Операційні об'єкти, тобто блоки, задають логіку функціонування моделі системи і визначають шлях руху транзактів між об'єктами апаратної категорії. Модель системи можна зобразити сукупністю блоків, об'єднаних відповідно до логіки роботи реальної системи у так звану *блок-схему*. Блок-схему моделі можна зобразити графічно, наочно показуючи взаємодію блоків у процесі моделювання.

У блоках можуть відбуватися події чотирьох головних типів:

- 1) створення або знищення транзактів;
- 2) зміна числового атрибуту об'єкта;
- 3) затримка транзакта на певний період часу;
- 4) зміна маршруту руху транзакта в моделі.

Версія GPSS, реалізована в системі GPSS World, містить 53 типи блоків. За призначенням блоки класифікують на кілька груп.

1. Блоки, які здійснюють модифікацію атрибутів транзактів:
 - а) генерування і знищення транзактів: GENERATE, SPLIT, TERMINATE, ASSEMBLE;
 - б) затримка у часі: ADVANCE;
 - в) синхронізація руху двох (MATCH) і кількох (GATHER) транзактів;
 - г) зміна параметрів транзактів: ASSIGN, INDEX, MARK, PLUS;
 - д) зміна пріоритету транзакта: PRIORITY.
2. Блоки, які змінюють послідовність руху транзактів (блоки передавання керування): DISPLACE, TRANSFER, LOOP, TEST, GATE.
3. Блоки, пов'язані з групуючою категорією: ADOPT, ALTER, EXAMINE, JOIN, REMOVE, SCAN.
4. Блоки, які описують об'єкти апаратної категорії:

- а) одноканальні пристрої: SEIZE, RELEASE, PREEMT, RETURN, FUNAVAIL, FAVAIL;
 - б) багатоканальні пристрої: ENTER, LEAVE, SAVAIL, SUNAVAIL;
 - в) ключі (логічні перемикачі): LOGIC.
5. Блоки, які зберігають необхідні значення для того, щоб використати їх пізніше: SAVEVALUE, MSAVEVALUE.
6. Блоки, які забезпечують отримання статистичних результатів:
- а) черги: QUEUE, DEPART;
 - б) таблиці: TABULATE.
7. Блоки для організації списку користувача: LINK, UNLINK.
8. Блоки для організації введення-виведення:
- а) відкриття/ закриття файлу: OPEN/ CLOSE;
 - б) зчитування/ запис у файл: READ/ WRITE;
 - г) встановлення позиції поточної стрічки: SEEK.
9. Спеціальні блоки: BUFFER, COUNT, EXECUTE, INTEGRATION, SELECT, TRACE, UNTRACE.

Об'єкти апаратної категорії – це абстрактні елементи, на які можна здійснити декомпозицію (поділ) устаткування реальної системи. Діючи на ці об'єкти, транзакти можуть змінювати їхній стан і впливати на рух інших транзактів. До об'єктів цього типу належать *одноканальні пристрої, багатоканальні пристрої та логічні ключі*.

Одноканальні пристрої (ОКП) – це устаткування, яке в будь-який момент часу може займати лише один транзакт. Наприклад, один канал зв'язку, каса продажу квитків з одним віконцем, один телефон-автомат тощо.

Багатоканальні пристрої (БКП) призначені для імітації устаткування, яке здійснює паралельне обслуговування, тобто може одночасно обслуговувати кілька транзактів. БКП можна використовувати як аналог, наприклад, багатоканальної ремонтної майстерні, квиткових кас з кількома віконцями, телефонного переговорного пункту з кількома кабінами тощо.

Інколи події, які відбувалися в системі раніше, можуть заблокувати, змінити рух транзактів і настання найближчих подій. Наприклад, якщо один канал зв'язку вийшов з ладу, то всі наступні замовлення на передавання повідомлень потрібно скерувати на інші канали зв'язку. Для моделювання таких ситуацій використовують *логічні ключі*. Транзакт може встановлювати ці ключі в позиції „увімкнено” або „вимкнено”. Пізніше стан ключів можуть перевіряти інші транзакти для визначення шляху їхнього руху.

Обчислювальна категорія слугує для опису таких ситуацій у процесі моделювання, коли зв'язки між компонентами системи найпростіше і компактно виражаються у вигляді математичних (аналітичних і логічних) співвідношень. З цією метою об'єктами обчислювальної категорії введені *арифметичні та булеві змінні і функції*.

Змінні – це складні вирази, які містять сталі, системні числові атрибути (СЧА), бібліотечні арифметичні функції, арифметичні та логічні операції.

Вирази можна застосовувати у змінних і в операторах GPSS. У першому випадку вирази визначають командами GPSS, у другому їх визначають як частину мови PLUS.

Кожному об'єкту відповідають атрибути, які описують його стан у певний момент часу. До них є доступ для використання протягом всього процесу моделювання; їх називають *системними числовими атрибутами*. Наприклад, об'єкт обчислювальної категорії генератор випадкових чисел має СЧА RN_n – число, обчислене генератором рівномірно розподілених випадкових чисел номер n ; об'єкт динамічної категорії транзакт має СЧА PR – пріоритет транзакта, що опрацьовують у даний момент. Всього в GPSS World є понад 50 СЧА.

Булеві змінні дають змогу користувачеві перевіряти в одному блоці GPSS одночасно кілька умов, враховуючи стану або значення цих умов та їхніх атрибутів, тобто в даному блоці здійснюється звертання до булевої змінної, вираз якої містить у собі перевірку кількох умов. Булеві змінні можуть мати вигляд комбінації стандартних числових атрибутів, пов'язаних між собою за допомогою булевих операторів, охоплюючи й інші змінні. Булеві змінні визначають так само, як і арифметичні, але замість арифметичних операцій перевіряються різноманітні логічні умови.

За допомогою *функцій* користувач може здійснювати обчислення неперервних або дискретних функціональних залежностей між аргументом функції (незалежна величина) і залежним значенням функції. Усі функції в GPSS задають табличним способом за допомогою команд опису функцій. Як і змінні, функції не зв'язані з певними блоками.

Крім бібліотечних арифметичних функцій, GPSS World має 24 *вмонтованих генератори випадкових чисел*.

Об'єкти запам'ятовуючої категорії забезпечують звертання до збережених значень. Комірки збережених величин і матриці комірок збережених величин використовують для зберігання деякої числової інформації. Наприклад, значення, занесене в комірку, може бути коефіцієнтом використання пристрою у певний момент часу. Будь-який активний транзакт здійснює запис інформації у ці об'єкти. Пізніше записану в ці об'єкти інформацію зчитує будь-який транзакт. Матриці мають до шести вимірів.

До *статистичних об'єктів* належать черги і таблиці. У будь-якій системі рух потоку транзактів може бути зупинений через недоступність пристроїв. Наприклад, потрібний ОКП може бути вже зайнятим або БКП, до якого треба увійти, вже заповнений. У цьому випадку затримані транзакти можуть бути поставлені в *чергу* – ще один тип об'єктів GPSS. Облік цих черг – одна з головних функцій планувальника. Планувальник автоматично накопичує певну статистику стосовно пристроїв і черг. Крім того, користувач може збирати додаткову статистичну інформацію, вказавши спеціальні точки в моделі.

Для полегшення табулювання статистичної інформації в GPSS передбачено спеціальний об'єкт – *таблицю*. Таблиці використовують для

отримання вибірових розподілів деяких випадкових величин. Таблицю складають з частотних класів (діапазонів значень), куди записують число, яке дорівнює кількості потраплянь конкретного числового атрибута до кожного, того чи іншого, частотного класу. Для кожної таблиці обчислюють також математичне сподівання і середнє квадратичне відхилення. Ця статистика є стандартною для всіх таблиць. Після завершення експерименту з моделлю результати, які містяться в таблицях, можуть виводитися або на дисплей або на друк.

До *групуючої категорії* належать три типи об'єктів: *числова група, група транзактів і списки*.

У деяких моделях транзакти є об'єктами, що характеризуються спільними СЧА, які зазвичай зображають як параметри транзакта і змінюються під час проходження транзактів через блок ASSIGN. Однак це обмежує доступ до атрибутів, а інколи доводиться одночасно змінювати деякі атрибути всіх транзактів цієї множини. Об'єкти такого типу дають змогу користувачеві звертатися до атрибутів транзактів даної *групи*. Вони працюють в одному з режимів: у режимі транзакта або числовому. Режим визначають першим звертанням до певної групи. У режимі транзакта величини, які надходять до групи, зображають номерами за чергою транзактів даної групи. У другому режимі величини, які надходять, зображають групою числових значень.

Під час моделювання транзакти зберігаються у списках. Є п'ять видів списків, проте лише в одному з них у будь-який момент часу може перебувати транзакт: 1) поточних подій; 2) майбутніх подій; 3) затримки ОКП або БКП; 4) відкладених переривань ОКП; 5) користувача.

Список поточних подій (СПП) охоплює транзакти, які відповідають подіям, час настання яких менший (транзакти, які повинні були почати рухатися в деякий момент у минулому, але були заблоковані) або дорівнює поточному часові. Транзакти з СПП мають увійти до блоків до чергової зміни модельного часу. У СПП транзакти розташовані у порядку спадання пріоритету. Транзакти з однаковими пріоритетами розташовуються в порядку їхнього надходження у список.

Список майбутніх подій (СМП) містить транзакти, які відповідають подіям, час настання яких більший від поточного часу, тобто подіям, які мають відбудися в майбутньому. Такі транзакти розташовуються в списку строго у порядку зростання часу початку руху. Пріоритети не впливають на порядок руху в цьому списку.

До *списку повторних спроб* належать транзакти, для яких не виконані умови входження до наступного блоку. Ці умови перевіряються під час спроби входження транзакта до блоків GATE, TEST, TRANSFER ALL і TRANSFER BOTH. Транзакти зі списку повторних спроб чекають на зміну СЧА. Після цієї зміни транзакт активується і перевіряються умови входження його до наступного блоку. Якщо умови виконуються, транзакт входить до наступного блоку і автоматично вилучається зі списку повторних спроб.

Одноканальний пристрій має:

- *список відкладених переривань* – список транзактів, які є в черзі до ОКП за пріоритетом;
- *список переривань* – список транзактів, обслуговування яких цим ОКП було перерване;
- *список затримки* – список транзактів, які перебувають в черзі до ОКП в порядку пріоритету;
- *список повторних спроб* – список транзактів, які очікують зміни стану ОКП.

Багатоканальний пристрій має:

- *список затримки* – список транзактів в порядку пріоритету, які очікують на можливість зайняти звільнені канали БКП;
- *список повторних спроб* – список транзактів, що очікують зміни стану БКП.

Список синхронізованих транзактів мають блоки ASSEMBLE, GATHER, MATH. Він містить транзакти, які перебувають в даний момент часу у стані порівняння.

Список користувача містить транзакти, вилучені користувачем зі списку поточних подій і введені до списку користувача як тимчасово неактивні. Списки користувача використовують для організації черг з усіма дисциплінами, крім дисципліни „першим прийшов – першим обслужений”.

2.2. Блок-діаграма. Структура моделі мовою GPSS

Опис моделі мовою GPSS здійснюють за допомогою блоків, які виконують певні операції в моделі. Набір блоків, з'єднаних між собою лініями, називають *блок-діаграмою*. Для побудови моделі з допустимої множини блоків вибирають необхідні і вибудовують з них діаграму так, щоб у процесі функціонування моделі вони взаємодіяли один з одним. Структура діаграми залежить від логічної схеми роботи реальної системи, яку моделюють на ЕОМ.

Конфігурація блок-діаграми моделі відображає напрями руху рухомих елементів – транзактів. Робота моделі полягає в переміщенні транзактів від блоку до блоку.

На початку моделювання в моделі немає жодного транзакта. У процесі моделювання транзакти входять до моделі в певні моменти часу відповідно до тих логічних потреб, які виникають у системі, яку ми моделюємо. Так само транзакти покидають модель у певні моменти часу залежно від специфіки моделювання.

Загалом у моделі є велика кількість транзактів, однак в певний момент часу рухається лише один транзакт. Він рухається від блоку до блоку шляхом, визначеним блок-діаграмою. Кожен блок можна розглядати як деяку точку, в якій відбувається звертання до підпрограми. У той момент, коли транзакт потрапляє до блоку, на виконання викликається відповідна підпрограма, і далі транзакт намагається увійти до наступного блоку. Таке пересування транзакта триває до того часу, поки не відбудеться одна з подій:

- транзакт потрапляє до блоку, функцією якого є затримка транзакта на деякий певний модельний час;
- транзакт потрапляє до блоку, функцією якого є вилучення транзакта з моделі;
- транзакт намагається увійти до наступного блоку відповідно до визначеної блок-діаграмою логіки, але блок відмовляється прийняти цей транзакт;
- транзакт залишається в тому блоці, в якому він в цей час перебуває, і буде повторювати свої спроби увійти до наступного блоку.

Коли умови в моделі зміняться, одна зі спроб може виявитися успішною і транзакт зможе продовжити свій рух у моделі.

Якщо виникла одна із зазначених умов, то транзакт залишається на місці і починається пересування в моделі іншого транзакта. Виконання моделі полягає у послідовному звертанні до підпрограм, які є наслідком входження в певні блоки транзактів, які рухаються. Проходячи через блоки, кожен транзакт вносить свій вклад у вміст *лічильників блоків*. Значення цих лічильників доступні розробникові через СЧА блоків: W – поточний вміст блоку і N – загальна кількість входжень до блоку.

Кожне просування транзакта в моделі є *подією*, яка повинна відбутися в певний момент модельного часу. Для того щоб підтримувати правильну послідовність подій у часі, планувальник має *таймер* модельного часу, який

автоматично коректується відповідно до логіки, визначеної моделлю.

Таймер GPSS має такі особливості:

- реєструються лише додатні дійсні значення часу;
- одиницю модельного часу (масштаб) визначає розробник моделі, який задає всі часові інтервали в одних і тих самих одиницях;
- планувальник не аналізує стан моделі в кожний наступний момент модельного часу (який відрізняється від поточного на одиницю модельного часу), а просуває таймер до моменту часу, коли повинна відбутися найближча наступна подія.

Значення таймера доступні розробникові через системні СЧА C1 (*відносний модельний час*) і AC1 (*абсолютний модельний час*).

Головним завданням, яке виконує планувальник, є визначення того, який транзакт потрібно вибрати наступним для просування в моделі, коли попередній транзакт припинив свій рух. З цією метою планувальник розглядає кожен транзакт як елемент деякого списку. Планувальник GPSS вводить транзакти залежно від умов в моделі до того чи іншого списку, проглядає списки, вибирає наступний транзакт для опрацювання, коректує таймер модельного часу після опрацювання всіх транзактів у списку поточних подій.

Змістовне значення транзактів визначає розробник моделі. Він встановлює аналогію між транзактами і реальними елементами системи, яку моделює. Таку аналогію ніколи не вказує планувальник GPSS. Вона присутня лише в пам'яті розробника моделі.

Після того як блок-діаграма побудована, вона повинна бути записана у формі, зручній для створення об'єкта "Модель". Для цього інформацію про об'єкти GPSS записують у вигляді послідовності операторів. Оператор моделі в GPSS World – це або оператор GPSS, або PLUS-оператор. Оператори GPSS – це або оператори блоків, які створюють блоки, або команди, які не створюють блоків.

На відміну від PLUS-операторів, які можуть містити кілька стрічок, оператори GPSS повинні записуватися одною текстовою стрічкою довжиною не більше 250 символів. Оператори GPSS складаються з частин, які називають полями. Поле – це набір символів, відокремлений пробілами або обмежувачем. Загалом оператор складається з таких полів: номер стрічки (необов'язково) | мітка (необов'язково) | дієслово (обов'язково) | операнди (залежить від дієслова) | коментар (необов'язково).

Важливо пам'ятати, що в GPSS *під час запису будь-якого оператора, який містить кілька операндів, відокремлених комами, після коми не може бути пробілу.*

Номери стрічок залишено для сумісності з GPSS/PC, їх ігнорує GPSS World. Оператори, незважаючи на встановлений для кожного формат, мають вільну форму запису в тому сенсі, що немає необхідності вирівнювати поля операторів за колонками.

Для того щоб розрізнити об'єкти GPSS, наприклад ОКП, БКП, черги тощо,

яких в моделі може бути кілька, їм дають імена. Імена мають починатися з літери, можуть містити до 200 літер і цифр, а також символи підкреслювання. Ім'я не повинне збігатися з ключовими словами GPSS. Одне і те ж саме ім'я можна використовувати для різних об'єктів.

У процесі моделювання можливі переходи від одного об'єкта моделі до іншого. Щоб це здійснити, об'єкт ідентифікують міткою, для чого служить поле мітки. Мітки утворюються так само, як імена блоків. GPSS присвоює міткам номери, починаючи з 10000. Користувач не може відразу записати замість мітки номер. Якщо він хоче це зробити, треба перед звертанням до об'єкта командою EQU присвоїти мітці номер, який не може бути більшим від 9999. Мітки також не можуть збігатися з ключовими словами GPSS.

У полі дієслова користувач має записати ключове слово GPSS, тобто або блок, або команду. Задання вхідних даних, необхідних для виконання операцій, які відповідають блокові або команді, здійснюють у полі операндів. Операнди відокремлюються комами. Поля операндів залежно від типу операції можуть містити до семи операндів, розташованих у певній послідовності, які позначають, зазвичай, першими літерами латинського алфавіту від А до G. Деякі оператори взагалі не мають операндів, а в деяких операнди опускають, автоматично встановлюючи їхні стандартні значення (за замовчуванням). Під час запису операндів використовують позиційний принцип: пропуск операнда відзначають комою.

Команди використовують для опису багатоканальних пристроїв, змінних, матриць комірок, функцій, таблиць і управління процесом моделювання.

2.3. Об'єкти обчислювальної категорії

Як уже було відзначено, до об'єктів обчислювальної категорії належать арифметичні і булеві змінні та функції. Змінна – це узагальнене поняття, яке в конкретному випадку є виразом. Вираз може охоплювати такі елементи:

- константи;
- системні числові атрибути;
- арифметичні операції;
- логічні операції;
- бібліотечні функції;
- змінні користувача.

Крім застосування в змінних, вирази в дужках можна також використовувати в операторах GPSS (деякі оператори допускають застосування і без дужок, але краще використовувати дужки).

Розглянемо елементи виразів.

2.3.1. Константи

Константи – це числа, які зображають в одному з трьох типів даних: цілочисельному, дійсному або рядковому. Оскільки арифметичні оператори перетворюють свої операнди в потрібну форму, то у виразах можна використовувати всі три типи даних.

Цілочисельні дані зберігаються у вигляді 32-розрядних чисел. Якщо відбувається переповнення, то ціле число перетворюється в дійсне.

Дійсні дані зберігаються у вигляді 64-розрядних цілих чисел з плаваючою крапкою подвійної точності. Мантиса може змінюватися від –308 до 308, а точність обмежена 15 десятковими розрядами.

Рядкові дані зберігаються в масивах символів ASCII будь-якої довжини. Довжина обмежена тільки значенням Max Memory Request (Максимальний запит пам'яті). Цю позицію можна змінити за допомогою меню Edit ►Settings ►Simulation.

2.3.2. Системні числові атрибути

У процесі моделювання GPSS World автоматично реєструє і коректує інформацію, подану системними числовими атрибутами різних об'єктів, які входять до моделі. Повний перелік СЧА, класифікованих за об'єктами GPSS, наведено в Додатку 2.

Імена СЧА зарезервовані. За *способом визначення імен СЧА* можна розбити на три групи.

Ім'я СЧА в *першій групі* складається з двох частин. *Перша частина* вказує групове ім'я, яке ідентифікує тип об'єкта і тип інформації про об'єкт. *Друга*

частина ідентифікує конкретного члена групи. *Групове ім'я* складається з однієї-двох літер, фіксованих для інформації про об'єкти певного типу, наприклад, Q – посилання на поточне значення довжини черги, FR – коефіцієнт використання пристрою, SR – коефіцієнт використання пам'яті тощо.

Об'єкти GPSS можуть бути ідентифіковані за допомогою числових або символічних імен. Ідентифікація СЧА залежить від вигляду адресації, яких в GPSS передбачено дві: пряма і непряма.

Розглянемо *пряму адресацію*. Якщо об'єкт ідентифіковано за допомогою номера, то посилання на його СЧА записується як

$$\text{СЧА}_j,$$

де j – номер об'єкта (додатне ціле число). Наприклад, Q4 – поточне значення довжини черги номер 4, FR3 – коефіцієнт використання пристрою номер 3. Якщо зазначають ім'я об'єкта, то посилання на його СЧА записується як

$$\text{СЧА}\$im'я,$$

де *ім'я* – ім'я об'єкта. Наприклад, Q\$Cherha – поточне значення довжини черги з іменем Cherha, SR\$Prystriy – коефіцієнт використання БКП з іменем Prystriy.

У разі непрямої адресації СЧА визначають як

$$\text{СЧА}*\text{параметр}.$$

Це означає:

- $\text{СЧА}*j$, j – номер параметра активного транзакта (додатне ціле число), яке містить номер потрібного блоку;
- $\text{СЧА}*im'я$, *ім'я* – ім'я параметра активного транзакта, яке містить номер потрібного блоку.

Наприклад, Q*4 – поточне значення довжини черги, номер якої є значенням параметра 4 активного транзакта, SR*Prystriy – коефіцієнт використання пам'яті, номер якої міститься в параметрі з іменем Prystriy активного транзакта.

Другу групу, як часковий випадок першої групи, становлять матричні СЧА. Матричний СЧА MX може містити до трьох ідентифікаторів (непрямих адрес). Наприклад:

$$\text{MX}*\text{Result}(*\text{Strichka},*\text{Stovpec})$$

є посиланням на матрицю, номер якої міститься в параметрі активного транзакта з іменем Result, а елемент матриці – номер стрічки і номер стовпця – визначають значеннями параметрів з іменами Strichka і Stovpec відповідно того самого активного транзакта.

Третю групу складають СЧА A1, AC1, C1, M1, PR, TG1, XN1, Z1, які називають атомарними. Ці СЧА, на відміну від СЧА першої і другої груп, не потребують зазначення номера або імені об'єкта.

СЧА за *ознакою доступності до них користувача* можна розділити на дві групи.

До першої групи належать атрибути, які в частині, що стосується їхнього формування, доступні лише GPSS World, однак їх може використати також і

користувач у виразах, але не може їх змінювати в процесі моделювання. Ця група містить найбільшу кількість СЧА.

Друга група охоплює СЧА обчислювальних об'єктів, які може змінювати розробник моделі і використовувати їх у виразах:

- FN_j (FN\$Rozpod) – обчислене дійсне значення функції номер j (з іменем Rozpod);
- V_j (V\$Imo) – обчислене значення дійсної або цілочисельної змінної номер j (з іменем Imo);
- BV_j (BV\$Zminna) – обчислене дійсне значення булевої змінної номер j (з іменем Zminna).

2.3.3. Арифметичні, умовні та логічні оператори

Арифметичні, умовні і логічні оператори призначені для позначення операцій, які виконують над елементами виразу. В GPSS World використовують такі оператори (табл. 2.2).

Таблиця 2.2

Оператор	Операція	Аргумент, значення
^	Піднесення до степеня	A^B ; A піднесене до степеня B
#	Множення	$A\#B$; множення A на B
/	Ділення	A/B ; частка від ділення A на B
\	Цілочисельне ділення	$A\backslash B$; частка від цілочисельного ділення A на B
@	Ціла остача	$A@B$; ціла остача від ділення A на B
-	Віднімання	$A-B$
+	Додавання	$A+B$
>= або 'GE'	Більше або дорівнює	1, якщо $A \geq B$, інакше – 0
<= або 'LE'	Менше або дорівнює	1, якщо $A \leq B$, інакше – 0
> або 'G'	Більше	1, якщо $A > B$, інакше – 0
< або 'L'	Менше	1, якщо $A < B$, інакше – 0
= або 'E'	Дорівнює	1, якщо $A = B$, інакше – 0
!= або 'NE'	Не дорівнює	1, якщо $A \neq B$, інакше – 0
& або 'AND'	Логічне „І”	1, якщо A і B не 0, інакше – 0
або 'OR'	Логічне „АБО”	1, якщо A або B або обидва не 0, інакше – 0

Якщо оператор потребує певного типу даних, то дані перетворюються автоматично. Наприклад, якщо над рядковою величиною виконується числова операція, то буде використаний числовий еквівалент цього рядка символів.

Оператор „#” використано для операції множення для того, щоб була можливість застосовувати оператор „*” для позначення непрямої адресації.

Однак користувач може змінювати значення операторів „#” і „*”. Для цього потрібно скористатися прапорцем Switch * and #, який можна знайти за допомогою меню Edit ►Settings ►Simulation.

Операції над елементами виразу виконуються в такій послідовності у порядку спадання пріоритетів:

- ^ – піднесення до степеня;
- #, /, \ – множення, ділення, цілочисельне ділення;
- @ – ціла остача;
- -, + – віднімання, додавання;
- >=, <=, >, < – оператори порівняння;
- =, != – дорівнює, не дорівнює;
- & – логічне „І”;
- | – логічне „АБО”.

Наприклад, вираз $2\#5^2+36$ означає $2 \cdot 5^2 + 36 = 86$. Цей вираз можна записати і в такому вигляді: $((2\#(5^2))+36)$. Щоб змінити встановлений порядок виконання арифметичних операцій, використовують круглі дужки. Наприклад, результат обчислення виразу $(2\#5)^2+36$ рівний $(2 \cdot 5)^2 + 36 = 100 + 36 = 136$.

2.3.4. Бібліотечні математичні функції

Бібліотека GPSS World має такі математичні функції (процедури) (табл. 2.3).

Таблица 2.3

Оператор	Функції
ABS(A)	Абсолютне значення
ATN(A)	Арктангенс
COS(A)	Косинус
EXP(A)	Експонента
INT(A)	Виділення цілої частини
LOG(A)	Натуральний логарифм
SIN(A)	Синус
SQR(A)	Квадратний корінь
TAN(A)	Тангенс

Аргументи бібліотечних функцій у всіх випадках автоматично перетворюються у числові значення. Допустимі значення аргументу – Expression (Вираз). Обчислювані числові значення функцій – дійсного типу.

2.3.5. Бібліотечні генератори випадкових чисел

Генератори випадкових чисел призначені для моделювання випадкових величин і випадкових подій.

Використовуючи табл. 2.4, варто наголосити на такому:

- всі аргументи генераторів мають допустиме значення Expression;
- після обчислення аргумент перетворюється до відповідного типу;
- символом „*” позначені аргументи, які автоматично перетворюються у ціле число;
- символом „**” позначені аргументи, які автоматично перетворюються у дійсне число;
- аргумент А після обчислення виразу і перетворення в ціле число використовують як номер генератора випадкових чисел;
- генерування випадкових чисел здійснюється за виразами.

Таблиця 2.4

Розподіл, оператор, тип	Функції
Бета Beta(A,B,C,D,E) Дійсний	A* – номер генератора випадкових чисел, $A \geq 1$; [B*,C*] – інтервал генерованих значень, $B > C$; D** і E** – перше і друге визначальні значення, які використовують для вибору із сім’ї форм. Beta(A,B,C,1,1)=Uniform(A,B,C), Beta(A,B,C,1,2) – лівий, Beta(A,B,C,2,1) – правий трикутники.
Біномний Binomial(A,B,C) Цілий	A* – номер генератора випадкових чисел, $A \geq 1$; B* – кількість незалежних випробувань Бернуллі, $B > 0$; C** – ймовірність успішних випробувань Бернуллі, $C \in [0,1]$.
Дискретно-рівномірний DUniform(A,B,C) Цілий	A* – номер генератора випадкових чисел, $A \geq 1$; [B*,C*] – інтервал генерованих значень, $B > C$.
Показниковий Exponential(A,B,C)	A* – номер генератора випадкових чисел, $A \geq 1$;

2.3. Об'єкти обчислювальної категорії

Дійсний	<p>V^{**} – величина зсуву, яка визначає місцезнаходження розподілу; C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$, $V+C$ – середнє значення, при $V=0$ C – середнє значення.</p>
Екстремального значення типу A ExtValA(A,B,C) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$; V^{**} – величина зсуву, яка визначає місцезнаходження розподілу; C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$.</p>
Геометричний Geometric(A,B) Цілий	<p>A^* – номер генератора випадкових чисел, $A \geq 1$; V^{**} – ймовірність успішних випробувань Бернуллі, $V \in [0,1]$.</p>
Гама Gamma(A,B,C,D) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$; V^{**} – величина зсуву, яка визначає місцезнаходження розподілу; C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$; D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$; $\text{Gamma}(A,B,C,1) = \text{Exponential}(A,B,C)$.</p>
Обернений Гаусса InvGauss(A,B,C,D) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$; V^{**} – величина зсуву, яка визначає місцезнаходження розподілу; C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$; D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
Обернений Вейбулла InvWeibull(A,B,C,D) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$; V^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p>

	<p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$;</p> <p>D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
Лапласа Laplace(A,B,C) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>V^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$.</p>
Логлапласів LogLaplace(A,B,C,D) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>V^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$;</p> <p>D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
Від'ємний біномний NegBinom(A,B,C) Цілий	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>V^* – кількість успішних випробувань Бернуллі, після якої повертається кількість неуспішних випробувань $V > 0$;</p> <p>C^{**} – ймовірність успішних випробувань Бернуллі, $C \in [0,1]$.</p>
Логістичний Logistic(A,B,C) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>V^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$.</p>
Логлогістичний LogLogis(A,B,C,D) Дійсний	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>V^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують</p>

	<p>для стискання або розтягування розподілу, $C > 0$;</p> <p>D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
<p>Логнормальний LogNormal(A,B,C,D) Дійсний</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>B^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$;</p> <p>D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
<p>Парето Pareto(A,B,C) Дійсний</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>B^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$.</p>
<p>Пірсона типу 5 Pearson5(A,B,C,D) Дійсний</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>B^{**} – величина зсуву, яка визначає місцезнаходження розподілу;</p> <p>C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$;</p> <p>D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$.</p>
<p>Нормальний Normal(A,B,C) Дійсний</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>B^{**} – середнє значення розподілу;</p> <p>C^{**} – стандартне відхилення розподілу, $C > 0$;</p>
<p>Пуассона Poisson(A,B) Цілий</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>B^{**} – середнє значення розподілу.</p>
<p>Пірсона типу 6 Pearson6(A,B,C,D,E) Дійсний</p>	<p>A^* – номер генератора випадкових чисел, $A \geq 1$;</p> <p>$[B^*, C^*]$ – інтервал генерованих</p>

	значень, $B < C$; D^{**} і E^{**} – перше і друге визначальні значення, які використовують для вибору із сім'ї форм.
Трикутний Triangular(A,B,C,D) Дійсний	A^* – номер генератора випадкових чисел, $A \geq 1$; $[B^*, C^*]$ – інтервал генерованих значень, $B < C$; D^{**} – мода розподілу.
Рівномірний Uniform(A,B,C) Дійсний	A^* – номер генератора випадкових чисел, $A \geq 1$; $[B^*, C^*]$ – інтервал генерованих значень, $B < C$; $\text{Uniform}(A,B,C) = \text{Beta}(A,B,C,1,1)$.
Вейбулла Weibull(A,B,C,D) Дійсний	A^* – номер генератора випадкових чисел, $A \geq 1$; B^{**} – величина зсуву, яка визначає місцезнаходження розподілу; C^{**} – величина, яку використовують для стискання або розтягування розподілу, $C > 0$; D^{**} – визначальне значення, яке використовують для вибору із сім'ї форм, $D > 0$; $\text{Weibull}(A,B,C,1) = \text{Exponential}(A,B,C)$; $\text{Weibull}(A,B,C,2)$ – розподіл Релея.

2.3.6. Змінні користувача

GPSS надає користувачеві можливість мати свої змінні для зберігання потрібних під час моделювання як числових, так і рядкових даних. Змінні користувача створюються за допомогою команди EQU або присвоюючих PLUS-процедур. Наприклад:

```
Imo      EQU  0.75
Strichka EQU  15
Stovpez  EQU  9
Name     EQU  "Name"
```

У наведених перших трьох прикладах змінним користувача Imo, Strichka, Stovpez присвоюються числові значення, а змінній Name – рядок.

За допомогою змінних користувача імена об'єктів GPSS можна замінювати номерами. Для цього необхідно раніше, як показано у прикладах, визначити

змінні користувача командою EQU.

Імена, задані командою EQU, які в моделі використовують як мітки об'єктів, вважають мітками.

Змінні користувача можна застосовувати в PLUS-виразах, змінювати командами EQU або присвоюючими операторами в PLUS-процедурах. Під час виконання експерименту з імітаційною моделлю чинники, вплив яких на шуканий показник передбачено дослідити, обов'язково мають бути змінними користувача.

2.3.7. Вирази в операторах GPSS

Вирази в системі GPSS World визначені як частина мови PLUS. Їх можна використовувати в операндах операторів GPSS і в PLUS-процедурах. В операндах, у яких допускається використання PLUS-виразів, їх беруть у дужки.

Отже, можна ввести чинник, будь-яке обчислення або виклик PLUS-процедури прямо в операнд. Наприклад:

```
ADVANCE (Imo1+(Exponential(2,0,Sered)))  
ASSIGN 2,((Normal(3,Sered,SerKilk)#1.75)
```

Крім того, за допомогою виразів, які містять СЧА, можна отримати доступ до змінних стану об'єктів GPSS, таких, як глобальні змінні користувача, з PLUS-процедур.

2.3.8. Комірки зберігання

В GPSS використовують комірки пам'яті, початкові значення яких можуть бути задані перед моделюванням і до яких можна звернутися з будь-якого місця моделі під час моделювання. Ці комірки є СЧА, їх називають *комірками зберігання*. Значення комірок зберігання змінюються лише за прямою вказівкою користувача. Перед початком моделювання встановлюють значення комірок зберігання, які дорівнюють нулю. За бажанням користувача окремим коміркам зберігання можуть бути присвоєні ненульові початкові значення за допомогою команди INITIAL. Команду INITIAL застосовують за відсутності активного транзакта або ж тоді, коли не можна використати значення параметрів транзактів. Вона має такий формат:

```
INITIAL A,[B]
```

Дужки [] означають, що цей операнд є необов'язковим. Операнд А під час ініціалізації комірки зберігання може бути *Хдодатна ціла*, *X\$ім'я*. Тобто в команді можна відразу зазначити або ім'я, або номер, хоч, як дізнаємось пізніше, не у всіх командах це можливо.

Операнд В – присвоюване початкове значення або UNSPECIFIED (не визначене). Може бути іменем, числом, рядком або UNSPECIFIED. Якщо операнд В не використовують, то значення комірки дорівнює 1.

Наприклад:

```
INITIAL X3,598.65
INITIAL X$Vyzn,Q$Cher
INITIAL X$Kom,Kor1
INITIAL X$Chas,"Rezult number ..."
INITIAL X2
```

У першому прикладі в комірку номер 3 заноситься число 598.65, у другому – в комірку з іменем Vyzn – поточна довжина черги з іменем Cher, у третьому – в комірку з іменем Kom – значення змінної корнистувача Kor1, у четвертому – в комірку з іменем Chas – заноситься рядкова константа, а в п'ятому – операнд В не використовується, тому комірці зберігання номер 2 присвоюється значення 1.

У процесі моделювання значення комірки зберігання змінюється з входженням транзакта до блоку SAVEVALUE (зберегти величину). Формат запису блоку SAVEVALUE:

```
SAVEVALUE A,B
```

Коли транзакт потрапляє до блоку SAVEVALUE, то величина, яка є в операнді В, стає значенням комірки зберігання, номер або символічне ім'я комірки якої записане в операнді А. Наприклад, відразу після входження транзакта до блоку

```
SAVEVALUE P4,V$Zmin
```

обчислюється значення змінної Zmin. Отриманий результат призначається комірці зберігання, номер якої записаний в параметрі номер 4 активного транзакта. Одночасно старе значення комірки зберігання знищується.

Операнд А може бути іменем, додатним цілим числом, виразом у дужках, СЧА*параметром або СЧА, а операнд В, крім того, може бути числом або рядком.

Блок SAVEVALUE може бути використаний як у режимі заміщення, так і в режимі накопичення або зменшення. У режимі накопичення попереднє значення комірки зберігання збільшується на значення, яке є у полі операнда В.

У режимі зменшення воно зменшується на величину, яка зазначена в операнді В. Режими накопичення або зменшення визначаються введенням відповідно знаку + або – перед комою, яка відокремлює операнди А і В. Наприклад:

```
SAVEVALUE 4+,X3
SAVEVALUE Dva-,V$Imo
SAVEVALUE Tret,-18
SAVEVALUE 4, (5.57#X$Nom2+Q3)
SAVEVALUE 15,"Rezult number ..."
```

Значення операндів А змінюються лише із входженням транзактів до цих блоків. У першому прикладі значення комірки зберігання номер 4 з входженням транзакта до блоку SAVEVALUE збільшується на значення комірки зберігання з номером 3. У другому прикладі значення комірки

зберігання з іменем *Dva* зменшується на обчислене значення арифметичної змінної *Imo*. У третьому прикладі значення комірки зберігання з іменем *Tret* заміщується числом -18 . У четвертому прикладі обчислюється вираз у дужках і присвоюється комірці номер 4. У п'ятому вміст комірки номер 15 заміщується рядком. Якщо комірки з таким іменем не існує, то вона створюється.

Комірки зберігання мають єдиний СЧА з назвою *X*, значенням якого є поточне значення відповідної комірки зберігання. СЧА *X* в операнді *A* блоку *SAVEVALUE*, як видно з прикладів, не використовується. Під час запису інших операндів посилання на СЧА *X* обов'язкове. Посилання на цей СЧА застосовують також у виразах, функціях, в команді *INITIAL*.

2.3.9. Матриці комірок зберігання

Поняття комірки зберігання в GPSS застосовують і до матриць. Ці об'єкти дають змогу впорядкувати значення, які потрібно зберегти у вигляді *матриць*. Матрицю необхідно спочатку описати. Команда опису матриці має такий формат:

```
Name MATRIX A,B,C,[D],[E],[F],[G]
```

Мітка *Name* визначає ім'я матриці і повинна бути іменем (не може бути числом). Операнд *A* не використовують, його залишено для сумісності зі старішими версіями GPSS. Операнди *B*, *C*, *D*, *E*, *F*, *G* можуть бути лише цілими додатними числами.

Операндом *B* задають кількість рядків матриці – максимальну кількість елементів у першому вимірі, операндом *C* – кількість стовпців матриці – максимальну кількість елементів у другому вимірі. Операнди *D*, *E*, *F*, *G* задають максимальну кількість елементів у третьому, четвертому, п'ятому і шостому вимірах відповідно. Наприклад, команда

```
Kor MATRIX ,3,5
```

визначає матрицю з іменем *Kor*, яка містить три рядки і п'ять стовпців.

Матриці мають єдиний СЧА з назвою *MX*, за допомогою якого можна звертатися до будь-якого елемента матриці. Значеннями рядків і стовпців можуть бути лише імена, цілі числа і параметри транзактів. Наприклад:

```
MX1(2,5)  
MX$Tab(P3,P$Imo)  
MX$Tab(Viser,P4)
```

У першому прикладі визначається елемент матриці номер 1, який перебуває на перетині другого рядка і п'ятого стовпця. У другому прикладі для визначення елемента матриці з іменем *Tab* використовуються значення параметра номер 3 і параметра з іменем *Imo* транзакта, а у третьому – ім'я змінної користувача і параметр номер 4 транзакта. Змінній користувача раніше має бути присвоєне ціле додатне число командою *EQU*.

Потрібно звернути увагу на те, що у першому прикладі матриця має не ім'я, а номер, хоч у команді визначення матриці, як було зазначено вище, на

місці мітки Name можна використовувати лише ім'я. Для присвоєння матриці номера замість імені потрібно для запису команди визначення матриці використати команду EQU. Наприклад:

```
Rozr EQU 7
Rozr MATRIX ,3,5
```

У цьому прикладі ім'я Rozr матриці замінене на номер 7, за яким (MX7) потрібно звертатися до матриці. Якщо в команді MATRIX замість імені Rozr відразу записати число 7, то під час трансляції виникне зупинка із зазначенням помилки “Пропущена мітка оператора”.

Перед початком моделювання або виконання команди CLEAR ON значення всіх елементів матриці мають дорівнювати нулеві. Деяким або всім елементам за бажанням розробника можна присвоїти значення, які не дорівнюють нулю, або вони можуть бути переведені в невизначений стан командою INITIAL. Наприклад:

```
INITIAL MX3(2,4),12.37
INITIAL MX$Ktp(3,8),75
INITIAL MX$Kran(2,3),Kontr
INITIAL Petit,12
INITIAL Zsuv
INITIAL Mist,UNSPECIFIED
```

У першому прикладі елементові матриці номер 3, який розташований на перетині другого рядка і четвертого стовпця, присвоюється початкове значення 12.37, у другому – елементові (3,8) матриці з іменем Ktp – значення 75, у третьому – елементові (2,3) матриці з іменем Kran – значення змінної користувача Kontr. У четвертому прикладі замість операнда зазначено ім'я матриці Petit, тому всім елементам цієї матриці присвоюється значення 12. У п'ятому прикладі також замість операнда А стоїть ім'я матриці Zsuv, але операнд В не використовується, тому всім елементам матриці присвоюється значення 1. У шостому прикладі замість операнда В стоїть ключове слово UNSPECIFIED, тому всі елементи раніше створеної матриці Mist будуть переведені в невизначений стан. Цим переведенням матриця готується для використання в експерименті, в якому можуть бути пропущені, тобто не отримані під час експерименту дані. Невизначений стан елемента матриці засвідчує те, що даних немає. Якщо ж елементи матриці були б переведені в нуль, то це б сприймалось як отримання даних експерименту, які дорівнюють нулеві, що не відповідає дійсності.

Після експерименту матриця з його результатами передається бібліотечній процедурі дисперсійного аналізу (ANOVA), яка опрацьовує елементи, що перебувають в стані UNSPECIFIED, як пропущені дані замість нульових значень.

Оператор MATRIX створює матрицю в поточній моделі. Матриця не може бути вилучена з поточної моделі. Якщо оператор MATRIX вилучається з моделі, яка є у процесі виконання, то зв'язок з матрицею в цій моделі

залишається. Матрицю можна перевизначити повторно іншим оператором MATRIX з тою самою міткою. Перед використанням матриця має бути визначена в операторі MATRIX. Матриця обмежена максимальним обсягом пам'яті, встановленим у журналі налаштувань моделі.

Для запису у процесі моделювання значень у матриці, а також для збільшення або зменшення значень, записаних у матриці, слугує блок MSAVEVALUE. Формат його запису такий:

MSAVEVALUE A,B,C,D

В операнді А задають ім'я або номер матриці. Крайнім правим символом операнда може бути знак + (режим накопичення) або знак – (режим віднімання). В операнді В задають номер рядка, в операнді С – номер стовпця матриці. Операнд D визначає значення, яке має зберігатися, додаватися або відніматися. Всі операнди є обов'язковими. Вони можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

```
MSAVEVALUE 1,5,P3,135
MSAVEVALUE X$Abr-,3,7,V$Ser
MSAVEVALUE Krab+,4,(lmo#2+7),M1
MSAVEVALUE (Kra+Q$Che)+,(Roz-7),4,P$Od
MSAVEVALUE 3,Riad,Stovp,"Dovzhyna"
```

У першому прикладі з входженням транзакта до блоку MSAVEVALUE значення елемента матриці номер 1, розташованого на пертині рядка 5 і стовпця, номер якого міститься в параметрі 3 цього транзакта, заміщається числом 135. У другому прикладі значення елемента на перетині рядка 3 і стовпця 7 матриці, номер якої міститься у комірці зберігання Abr, зменшується на обчислене значення арифметичної змінної Ser.

У третьому прикладі номер стовпця знаходять як результат обчислення виразу в дужках (lmo#2+7), а потім значення визначеного елемента матриці Krab збільшується на величину відносного часу M1 перебування транзакта в моделі.

У четвертому прикладі номер матриці і номер рядка елемента визначають як результати обчислення виразів в дужках (Kra+Q\$Che) і (Roz-7) відповідно. Після цього значення знайденого елемента збільшується на величину, яка міститься в параметрі з іменем Od активного транзакта.

У п'ятому прикладі елементам матриці номер 3, розташованому на перетині рядка і стовпця, які визначають змінними користувача Riad і Stovp, присвоюється рядок "Dovzhyna". Якщо змінним користувача раніше командою EQU не будуть присвоєні відповідні значення, то виникне зупинка внаслідок помилки.

Зупинка внаслідок помилки виникає також в тому випадку, коли із входом транзакта до блоку MSAVEVALUE не буде знайдено матриці з потрібним іменем або номером, або не буде знайдений елемент з відповідними номерами рядка та стовпця.

СЧА MX у блоці MSAVEVALUE під час запису операнда A, як і СЧА X у блоці SAVEVALUE під час запису операнда A, не використовується. Посилання на СЧА MX застосовується у виразах, функціях, операторі INITIAL.

З розглянутого формату команди опису матриці випливає, що матриця GPSS може мати до шести вимірів. Однак, як видно з формату блоку MSAVEVALUE, він дає змогу викорисовувати лише двовимірні матриці, тобто в ньому доступні лише перші два виміри. Решта індексів дорівнюють одиниці.

Якщо є необхідність використання матриць більшої розмірності, то потрібно створити одну або кілька PLUS-процедур. PLUS-процедури мають доступ до всіх елементів всіх матриць. Матриці, визначені командою MATRIX, є глобальними і доступні всім PLUS-процедурам. На час виконання PLUS-процедури можуть бути створені тимчасові матриці з локальною областю видимості.

2.3.10. Арифметичні змінні й арифметичні вирази

Арифметична змінна є СЧА, який визначає користувач.

Якщо моделювання виконується не в режимі сумісності з GPSS/PC, то арифметична змінна в GPSS World може бути визначена командою VARIABLE або командою FVARIABLE, оскільки дія цих команд, а отже, й обчислювані значення, однакові. Формати команд такі:

```
Name VARIABLE A
```

```
Name FVARIABLE A
```

Тут операнд A – це вираз, який задає користувач, Name – ім'я арифметичної змінної.

Посиланням на арифметичну змінну є або V\$ім'я, або Vj, де j – номер змінної. Символічне ім'я можна замінити номером – додатним цілим числом, для чого треба використати команду EQU. Наприклад:

```
Voda EQU 3
```

```
Voda VARIABLE X$23/MX$Chas(1,P3)
```

Тепер на арифметичну змінну Voda можна посилатися так: V3. Команди EQU треба розміщати на початку моделі. У команді визначення змінної, так само як і в команді визначення матриці, замість імені відразу вказувати номер не можна, інакше на етапі трансляції станеться зупинка внаслідок помилки „Пропущено мітку оператора”.

Величиною арифметичної змінної є величина заданого користувачем арифметичного виразу. Значення арифметичної змінної можна використовувати так:

- як операнд; у цьому випадку значення арифметичної змінної може презентувати: номер об'єкта j; j логічного атрибуту (блок GATE); номер параметра транзакта (блоки ASSIGN, INDEX, LOOP, MARK, SPLIT); значення атрибуту;
- як аргумент функції;

- як значення залежної змінної атрибутивної функції;
- як аргумент таблиці;
- як операнд іншої арифметичної змінної або булевої змінної.

Наведемо приклади арифметичних виразів.

```
Bak VARIABLE R$Prys+S$Prys
Kas VARIABLE Q$Cher-Q$Bul
Dan VARIABLE P3+Q$Box/2
Las VARIABLE N$Ton@10
```

У першому прикладі змінна *Bak* є величиною, яка дорівнює сумі вільної ємності і поточного вмісту БКП *Prys*. Цей приклад показує, що розробник може задавати арифметичну змінну, величина якої є ємністю пристрою. У другому прикладі змінну *Kas* визначають як поточну довжину черги *Cher* мінус поточна довжина черги *Bul*. У третьому прикладі змінну *Dan* підраховують спочатку діленням поточної довжини черги *Box* на 2, а потім додаванням результату до значення параметра 3 активного транзакта. У четвертому прикладі *Las* – ціла остача від ділення на 10 кількості входжень до блоку з іменем *Top*.

Значення арифметичного виразу обчислюється, коли активний транзакт потрапляє до блоку, оператор якого серед своїх операндів містить одне або більше посилань на арифметичні змінні. Підраховані значення є дійсними числами.

В GPSS World проміжні результати і значення СЧА не округлюються. Округлення може задавати лише сам користувач. Наприклад:

```
Zmin1 VARIABLE MX$Matr(P$Strich,P$Stov)/Koryst
```

Підраховане значення змінної *Zmin1* буде цілим числом, оскільки використано операцію ділення націло. Для округлення можна використати і бібліотечну функцію *INT(A)* виділення цілої частини. Наприклад:

```
Zmin1 VARIABLE INT(MX$Matr(P$Strich,P$Stov)/Koryst)
```

Для ділення у цьому випадку використано звичайну операцію.

Якщо ж процес моделювання виконується в режимі сумісності з GPSS/PC, то обчислення одного і того ж арифметичного виразу, заданого командами *VARIABLE* і *FVARIABLE*, здійснюється по-різному, оскільки в GPSS/PC значення всіх СЧА є цілого типу.

У режимі сумісності під час опису арифметичної змінної командою *VARIABLE* всі проміжні обчислення і кінцевий результат округлюються до цілого значення. Дійсні змінні (з плаваючою крапкою) визначає команда *FVARIABLE*. У цьому випадку округлюється до цілого значення лише кінцевий результат, тобто дійсні змінні є також цілочисельні.

Арифметична змінна і змінна з плаваючою крапкою в режимі сумісності з GPSS/PC не можуть мати однакових імен. Якщо вони мають однакові імена, то під час обчислення використовують останній з цих двох описів. Це відбувається тому, що СЧА *V* використовують як для звертання до арифметичних змінних, так і до змінних з плаваючою крапкою. Спосіб обчислення змінної *V* є оператором опису цієї змінної.

Розходження між результатами, отриманими під час обчислення змінних з плаваючою і фіксованою крапками, можна побачити з наведених нижче прикладів:

```
Obch1  FVARIABLE  30#(7/5)
Obch2  VARIABLE   30#(7/5)
```

Значення змінної Obch1 буде дорівнювати 42, оскільки сталу 30 потрібно помножити на 1,4 і від результату 42 взяти цілу частину. Змінна Obch2 дорівнює 30, оскільки результат проміжної операції 1,4 буде округлений до 1.

Оскільки GPSS World надає можливість використовувати вирази в операндах, то доцільно для спрощення моделей вираз, який неодноразово використовують в різних частинах моделі, визначити як змінну лише один раз, а потім посилатися на нього.

2.3.11. Булеві змінні

Булеві змінні дають змогу, використовуючи лише один блок приймати рішення, зумовлені станом і значенням атрибутів багатьох об'єктів GPSS. Булева змінна є логічним виразом, складеним з різних СЧА, в тім числі й із інших булевих змінних. Вираз може також містити бібліотечні функції і Plus-процедури.

Булеву змінну визначають командою BVARIABLE. Формат команди такий:

```
Name  BVARIABLE  A
```

Тут Name – ім'я булевої змінної, яке так само, як ім'я арифметичної змінної, за бажанням користувача, командою EQU можна замінити номером. Операнд A – логічний вираз.

У виразі можна використовувати арифметичні й логічні оператори (див. табл. 2.2). Кінцевий результат перетворюється в ціле значення 0, якщо дорівнює нулю, або в ціле значення 1, якщо відмінний від нуля.

Логічні оператори зв'язані з об'єктами апаратної категорії, їх використовують для визначення цих об'єктів. Існують такі логічні оператори:

- FVj дорівнює 1, якщо пристрій j доступний, інакше – 0;
- FIj дорівнює 1, якщо пристрій j обслуговує переривання, інакше – 0;
- SFj дорівнює 1, якщо БКП j заповнений цілковито, інакше – 0;
- SEj дорівнює 1, якщо БКП j порожній, інакше – 0;
- SVj дорівнює 1, якщо БКП j доступний, інакше – 0;
- LSj дорівнює 1, якщо логічний ключ j увімкнутий, інакше – 0.

Під j розуміють номер або ім'я. Наприклад:

```
Stan1  BVARIABLE  FV$Kovp1
Stan2  BVARIABLE  SF$Kovp2
Stan3  BVARIABLE  SV$Kovp3
Stan4  BVARIABLE  LS4
```

У першому прикладі булева змінна Stan1 дорівнює 1, якщо пристрій Kopr1 доступний, і – 0, якщо недоступний. У другому прикладі булева змінна Stan2 дорівнює 1, якщо БКП Kopr2 цілковито заповнений. У третьому прикладі булева змінна Stan3 дорівнює 1, якщо БКП Kopr3 доступний. У четвертому прикладі булева змінна Stan4 дорівнює 1, якщо логічний ключ номер 4 увімкнений.

Оператори відношення (див. табл. 2.2) здійснюють алгебричне порівняння операндів. Наприклад:

```
Vidn1  BVARIABLE  V$Vsi 'G' 5
Vidn2  BVARIABLE  Q4 'LE' P2
Vidn3  BVARIABLE  MX$Matr(Str,Stov) 'GE' P3
```

Булева змінна Vidn1 дорівнює 1, якщо змінна Vsi більша 5, інакше дорівнює 0. У другому прикладі булева змінна Vidn2 дорівнює 1, якщо поточна довжина черги номер 4 не перевищує параметра 2 транзакта. У третьому прикладі булева змінна Vidn3 дорівнює 1, якщо значення певного елемента матриці Matr не менше ніж значення параметра 3.

Булевих операторів є два: OR – АБО і AND – І. Оператор АБО перевіряє виконання хоча б однієї з умов, а оператор І потребує виконання обидвох умов. Наприклад:

```
Por1  BVARIABLE  FI$Ves 'OR' SF5
Por2  BVARIABLE  FI$Ves 'AND' SF$Gros
Por3  BVARIABLE  (V2'G'5)'AND'(FN$Kop'OR' LS4)
```

Булева змінна Por1 дорівнює 1, якщо виконується одна з умов: пристрій Ves обслуговує переривання або БКП номер 5 незаповнений. Булева змінна Por2 дорівнює 1, якщо виконуються обидві умови: пристрій Ves обслуговує переривання і БКП з іменем Gros незаповнений. У третьому прикладі змінна Por3 дорівнює 1, якщо виконуються обидві умови: значення змінної номер 2 більше ніж 5 і логічний ключ номер 4 увімкнений.

Дужки у третьому прикладі потрібні лише для задання певних булевих співвідношень. Дужки треба використовувати лише в тих випадках, коли вони необхідні.

Якщо булева змінна задається СЧА, як наприклад:

```
Zmin  BVARIABLE  V$Avto
```

то обчислюється значення арифметичної змінної на ім'я Avto, і якщо воно відмінне від нуля, то значення булевої змінної Zmin буде дорівнювати 1, у протилежному випадку – 0.

Ми розглянули всі елементи виразів, крім функцій. Методи застосування функцій розглядатимемо в наступному розділі.

3. ПОБУДОВА МОДЕЛЕЙ З ОДНОКАНАЛЬНИМИ ТА БАГАТОКАНАЛЬНИМИ ПРИСТРОЯМИ

Для здійснення обслуговування використовують певні елементи, які в GPSS називають об'єктами апаратної категорії. До них належать одноканальні (ОКП) і багатоканальні пристрої (БКП) та логічні ключі (перемикачі).

ОКП характеризується двома головними властивостями:

- кожен ОКП 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, якщо він має вищий пріоритет, перервати обслуговування, яке відбувається в даний момент;
- коли на ОКП надходить транзакт, у моделі потрібно передбачити час, необхідний для його обслуговування.

Оскільки ОКП у моделі може бути багато, то для їхньої ідентифікації їм присвоюють символічні імена. Імена утворюються так само, як імена блоків.

Під час моделювання можливі такі режими організації функціонування ОКП:

- займання ОКП і його звільнення;
- переривання обслуговування ОКП;
- недоступність ОКП і відновлення доступності.

Оскільки потоки, які існують у реальних системах, у моделях імітують за допомогою транзактів, то перед тим, як розглядати блоки, які моделюють ОКП, дізнаємось, як вводяться транзакти до моделі і як виводяться з неї. У цьому розділі також розглянемо деякі блоки GPSS, без яких неможлива побудова найпростіших моделей.

3.1. Організація надходження транзактів до моделі й вилучення транзактів з неї

3.1.1. Надходження транзактів до моделі

GENERATE – це блок, через який транзакти заходять до моделі. На кількість різних блоків GENERATE в одній моделі обмежень не існує. Блок GENERATE має такий формат запису:

GENERATE [A],[B],[C],[D],[E]

Як і раніше, квадратні дужки означають, що даний операнд використовувати не обов'язково.

Інтервал часу між послідовними появами транзактів блоку GENERATE називають *інтервалом надходження*. 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*.

Операнд А – середній інтервал часу між послідовними надходженнями транзактів до моделі.

Операнд В задає модифікатор, який 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 транзакту вибирається число з отриманого інтервалу. Це буде значенням часу, через який наступний транзакт вийде з блоку GENERATE.

Наприклад, блок

```
GENERATE 10,4
```

створює транзакти через випадкові інтервали часу, рівномірно розподілені на відрізок [6; 14].

Можна вибрати генератор для розігрування рівномірно розподілених випадкових чисел. Для цього треба скористатися меню Edit ►Settings, де на сторінці Random Numbers (Випадкові числа) у полі блоку GENERATE потрібно ввести номер генератора – будь-яке додатне число. За замовчуванням використовується генератор номер 1.

Операнди А і В задавати не обов'язково. Коли один або обидва операнди не зазначені, то за замовчуванням припускається їхнє нульове значення. Наприклад, кожен з блоків

```
GENERATE 10.3,0
```

```
GENERATE 10.3
```

задає детерміноване значення 10,3 інтервалу надходження транзактів.

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

За будь-якого способу обчислення інтервалу часу значення операнда В не повинне перевищувати значення операнда А, в протилежному випадку у блоці GENERATE буде отримано від'ємний інтервал часу, який спричинить зупинку внаслідок помилки “Від'ємний час затримки”.

Розглянемо три додаткові операнди: С – зміщення інтервалів; D – обмежувач, E – рівень пріоритету.

Зміщення інтервалів (початкова затримка) С – це момент часу появи першого транзакта у блоці GENERATE. Після цієї першої появи всі наступні появи транзактів виникають згідно з розподілом інтервалів часу, які задають операнди 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 задаватися так само, як і операнди А та В: іменем, додатним числом, виразом у дужках або безпосередньо СЧА.

Операнд D задає граничне значення загальної кількості транзактів, які можуть увійти до моделі 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 до моделі через блок GENERATE. Для задання 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 можна задавати так само, як і операнди А, В і С, 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 А і D. *Не можна використовувати на полі операнда параметри транзактів.* Необхідно також пам'ятати, що транзакт не повинен заходити до блоку GENERATE. Якщо транзакт намагається це робити, то виникає помилка виконання.

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.

Наведемо приклади запису блоків GENERATE:

а) з операндом А

```
GENERATE 15.8
GENERATE X$Berta
GENERATE MX$Chas(2,3)
GENERATE V$Stan
GENERATE (Exponential(7,0,X$Mod))
GENERATE Berta
```

б) з операндами А і В

```
GENERATE 18.8,X$Vidn
GENERATE X$Sered,FN3
```

3.1. Організація надходження транзактів до моделі й вилучення транзактів з неї

GENERATE Serednie, FN5
GENERATE (V\$Konst+3.2), FN\$Met

в) з операндами А і С

GENERATE 5.3,,7.9
GENERATE 1.5,,X\$ChasNadh
GENERATE V\$IntNadh,,MX1(X\$Str,X\$Stov)
GENERATE (Normal(2,X\$Ser,X\$SerVymk),,Ste

г) з операндами А, В, Е

GENERATE 12.5,3.9,,7
GENERATE (Normal(2,X\$Ser,X\$SerVymk),Bert,,2
GENERATE V\$Ser,(V2-40.5),,,1

Наведені приклади показують різні способи задання операндів блоку GENERATE.

3.1.2. Генерування випадкових чисел в GPSS

Загалом джерелом „випадковості” в моделях GPSS є одна або кілька функцій, звертаючись до яких можна отримати випадкові значення з вибірки рівномірно розподілених на проміжку [0,1] чисел. Розігрування значення відповідно до деякого розподілу, зазвичай, здійснюють двома етапами:

- розігрується випадкове число з вибірки чисел, рівномірно розподілених на проміжку [0, 1];
- отримане число певним способом перетворюється на еквівалентне йому значення, яке вже належить до вибірки чисел потрібного нам розподілу.

В GPSS World передбачена необмежена кількість генераторів рівномірно розподілених псевдовипадкових чисел на проміжку [0,1]. Звертаються до генератора без попереднього його оголошення за допомогою СЧА RNn, де n – номер генератора. Початкове згенероване генератором число збігається з номером n генератора. Наприклад, запис RN823 означає, що генератор номер 823 запускається початковим числом 823.

Під час моделювання може виникнути необхідність дослідження:

- різних варіантів моделі із заданням одного і того ж самого вхідного потоку випадкових чисел;
- одного або кількох варіантів моделі із заданням різних потоків випадкових чисел.

Перша необхідність легко реалізується, оскільки генератори дають змогу відтворювати одні й ті ж самі послідовності рівномірно розподілених випадкових чисел.

У другому випадку GPSS надає можливість змінювати початкові числа перших семи RN1,...,RN7 генераторів і, як наслідок, формувати різні послідовності. Для цього призначена команда RMULT такого формату:

RMULT [A],[B],[C],[D],[E],[F],[G]

Операнд А – початкове число для RN1, В – для RN2, ..., G – для RN7.

Наприклад, запис

`RMULT ,23,74,,59`

означає встановлення початкових чисел для генераторів 2, 3 і 6. Початкові значення генераторів 1, 4, 5 і 7 залишаються без змін.

В GPSS World псевдовипадкові числа генеруються за 32-бітним мультиплікативним конгруентним алгоритмом, який дає змогу отримати 2147483646 унікальних псевдовипадкових чисел, тобто з періодом $2^{31}-2$, не враховуючи 0.

СЧА RN набуває цілих значень від 0 до 999 включно, а під час обчислення випадкових функцій використовується випадкове число з інтервалу від 0 до 0,999999 включно.

В GPSS World генератори випадкових чисел використовують не лише у блоках GENERATE і ADVANCE, а й під час опрацювання одночасних подій та у блоках TRANSFER, які працюють в режимі статистичної передачі. Користувач може вибрати будь-який генератор, знайшовши в меню Edit ►Settings на сторінці Random Numbers поле відповідного блоку (за замовчуванням – RN1).

Генерування випадкових чисел в GPSS World як для рівномірного, так і для інших типових розподілів найзручніше здійснювати за допомогою вмонтованих бібліотечних генераторів випадкових чисел. Однак з цією метою можна також використовувати й функції, які в GPSS призначені для обчислення величин, заданих табличними залежностями.

3.1.3. Особливості застосування функцій

Для застосування в GPSS функція має бути задана таблицею. Правила пошуку значення функції за таблицею визначає команда FUNCTION, після якої йдуть пари можливих значень аргумента й функції. Команда FUNCTION має такий формат:

`Name FUNCTION A,B`

Тут Name – ім'я функції. Операнд A – аргумент функції, який може бути іменем, додатним цілим числом, рядком, виразом у дужках, СЧА або СЧА*параметром.

Операнд B складається з однієї літери, яка визначає тип функції, й цілого додатного числа, яке задає кількість пар можливих значень аргумента й функції (кількість точок функції).

В GPSS World можна застосовувати такі п'ять типів функцій:

- C – неперервна числова;
- D – дискретна числова;
- E – дискретна атрибутивна;
- L – спискова числова;
- M – спискова атрибутивна.

Існує ще один тип функції, яку застосовують у блоках GENERATE й

ADVANCE на місці операнда В. Це – модифікатор-функція. Значення цієї функції обчислюється з подвійною точністю й множиться на значення операнда А. Отриманий результат використовують як потрібну часову затримку в цьому блоці.

Значення, задані командою FUNCTION, зберігаються у формі чисел з плаваючою крапкою подвійної точності. Ці значення обмежені 306 десятичними розрядами з точністю 15 десятичних розрядів.

Функції всіх типів мають єдиний СЧА з назвою FN, значенням якого є обчислене значення функції. Обчислення функції відбувається під час потрапляння транзакта до блоку, який містить посилання на СЧА FN з іменем функції.

Для деяких функцій зручніше використовувати не функції, а Plus-процедури. Однак функції GPSS краще підходять для застосування їх як табличних функцій. Крім того, функція GPSS може обчислюватися за менший час, ніж аналогічна Plus-процедура.

Під час визначення інтервалів надходження й обслуговування замовлень часто виникає необхідність розігрування випадкових чисел для дискретних розподілів, відмінних від рівномірних. Розглянемо особливості розігрування таких чисел з використанням функцій типу D.

Припустимо, що функція, яку потрібно використовувати в моделі, задана табл. 3.1. Згідно з цією таблицею випадкова величина повинна набувати значення 3, 6, 8, 9, 11 з частотою 0,14; 0,21; 0,23; 0,22; 0,2 відповідно.

Таблиця 3.1

Значення випадкової величини	Відносна частота	Накопичена частота	Діапазон	Інтервал
3	0,14	0,14	0,0...0,14	1
6	0,21	0,35	0,14...0,35	2
8	0,23	0,58	0,35...0,58	3
9	0,22	0,8	0,58...0,8	4
11	0,2	1,0	0,8...1,0	5

Розіграємо число на проміжку $[0, 1]$. Нехай це буде 0,533635. Воно потрапляє до інтервалу 3. Отже, шуканим значенням випадкової величини є 8.

Отже, коли необхідно розіграти випадкове число відповідно до деякого дискретного розподілу, планувальник повинен автоматично виконати процедуру перегляду таблиці, якою задана функція. Для цього йому потрібно мати:

- джерело випадкових чисел з відрізка $[0, 1]$;
- значення випадкової величини;
- накопичену частоту набування значень випадковою величиною.

Ці дані користувач задає за допомогою *функції типу D – дискретного*

tiny. У списку даних функції типу D значення аргумента X мають бути типу Expression, а значення функції Y – типу Integer, Real або Name. Для цього прикладу функція, задана табл. 3.1, буде визначена командою FUNCTION так:

```
Rozp FUNCTION RN3,D5
.14, 3/ .35, 6/ .58, 8/ .8, 9/ 1, 11
```

Функція має ім'я Rozp. Джерелом випадкових чисел є RN3. Випадкова величина може мати п'ять різних значень. Накопичені частоти і відповідні п'ять значень функції записані як п'ять пар чисел.

Під час написання пар чисел, які визначають координати X та Y точки не лише дискретної, а й функції будь-якого типу, потрібно дотримуватись таких правил:

- значення X та Y одної точки функції відділяють комою;
- набори координат відокремлюють знаком / або переходом до наступного рядка і розташовують у довільній кількості рядків;
- значення X розташовують у порядку зростання;
- кожна функція повинна мати хоча б дві описані точки;
- кількість пар даних, зазначених в операнді B команди FUNCTION, має збігатися з кількістю пар, відокремлених символом / у списку даних функції;
- списки даних функції не мають полів коментаря.

Функцію типу D можна використовувати у блоці GENERATE для генерування транзактів з дискретними довільними інтервалами. Наприклад:

```
DRozp FUNCTION RN43,D4
.14, 3.1/ .54, 4.6/ .74, 8.2/ 1, 1.9
GENERATE FN$DRozp
```

Тут блок GENERATE генерує транзакти з інтервалами 3,1; 4,6; 8,2 і 1,9 відповідно, ймовірності появи яких становлять 0,14; 0,4 (=0,54–0,14); 0,2 (=0,74–0,54) і 0,26 (=1–0,74) відповідно.

Так само функцію типу D можна використовувати у блоці ADVANCE, який ми розглянемо пізніше.

Для отримання випадкових чисел з неперервного рівномірного розподілу використовують *функції типу C*. Неперервну функцію визначають за допомогою символу C (замість D для дискретної функції), який записують на першому місці в операнді B. У списку даних функції змінні X та Y мають бути типу Integer або Real. Наприклад:

```
ChasNadh FUNCTION RN15,C4
.2, 15.3/ .32, 20.2/ 0.5, 13.4/ 1, 75
```

Простежимо відмінності між використанням неперервної та дискретної функцій. Спочатку неперервну функцію обчислюють так само, як і дискретну. Після звертання до функції відбувається розігрування випадкового числа, яке надалі використовують як аргумент функції. Потім переглядають таблицю для визначення інтервалу значень накопиченої частоти, до якого потрапило випадкове число. Якщо функція була визначена як дискретна, то беруть другий

елемент відповідної пари, його значення приймають за значення функції. Якщо ж функція визначена як неперервна, то виконується лінійна інтерполяція для пари точок X_i і X_{i+1} – кінців того інтервалу значень частоти, до якого потрапило випадкове число X , яке є аргументом. Інтерполяцію здійснюють згідно з формулою:

$$Y(X) = Y(X_i) + \frac{Y(X_{i+1}) - Y(X_i)}{X_{i+1} - X_i} \cdot (X - X_i), \quad i = \overline{1, m},$$

де m – кількість точок функції.

Неперервну функцію задають за допомогою лінійної інтерполяції, тому для всіх точок на одному інтервалі накопиченої частоти ймовірність їхнього використання є однаковою. Унаслідок цього процедура розіграшу неперервної рівномірно розподіленої випадкової величини є простою. Припустимо, що деяка випадкова величина рівномірно розподілена на проміжку $[3;7]$. Тоді ймовірність того, що значення змінної менше ніж 3, дорівнює 0, а ймовірність того, що вона менша ніж 7, дорівнює 1. Ці два значення ймовірності використовують як значення накопиченої частоти для визначення неперервної функції:

```
NeperRozp1 FUNCTION RN42,C2  
0, 3/ 1, 7
```

Незважаючи на те, що значенням другої пари, використаної для визначення функції, є 7, функція ніколи не набуває значення 7, тому що СЧА RN під час обчислення функції не може перевищувати 0,999999. Отож найбільшою цілою частиною отриманого значення є 6.

Перевага неперервної функції порівняно з дискретною стає очевидною, якщо результатом розігрування мають бути цілі числа. Припустимо, що необхідно розігрувати випадкові числа на проміжку від 200 до 585. Функцію можна описати так

```
NeperRozp2 FUNCTION RN34,C2  
0, 200/ 1, 586
```

Зауважимо, що друга пара тут 1, 586, а не 1, 585. Це зроблено для забезпечення можливості отримання максимального значення 585.

Якщо ж для розігрування випадкових чисел на тому самому проміжку використати дискретну функцію, то потрібно буде рахувати 386 пар накопичених частот і задати 386 пар для визначення функції.

Розглянемо функції типу E, L і M.

Як відомо, в *дискретній числовій функції типу D* значення аргументу X мають бути типу Expression, а значення функції Y – типу Integer, Real або Name. У дискретній атрибутивній функції типу E аргумент X також має бути типу Expression, а Y – типу Integer, Real, Name, СЧА або виразом у дужках. Наприклад:

```
Vyzn FUNCTION P2,E4  
1, V$User1/ 5, V$User2/ 10, R$Corp/ 12, FN$Chas
```

Функцію типу E обчислюють так само, як і функцію типу D. Відмінність

полягає лише в тому, що після ідентифікування аргументу X відбувається обчислення відповідного СЧА або виразу в дужках. Наприклад, для P2, який дорівнює 11, відбувається обчислення функції з іменем Chas.

Якщо у функції типу E значення Y є типу Integer, Real або Name, то вона ідентична функції типу D.

Якщо послідовність значень аргументу таблиці з координатами точок функції становлять числа натурального ряду 1, 2, 3,..., n, то таку дискретну функцію з метою економії пам'яті й машинного часу зручно визначити як *спискову числову функцію – функцію типу L*. У функції типу L аргумент X може бути лише типу Integer, його розглядають як порядковий номер. Значення функції Y – Integer, Real або Name. Наприклад:

```
Class FUNCTION Q$Dovzh,L4  
1, Stan1/ 2, Stan2/ 3, Stan3/ 4, Stan4
```

За порядковим номером знаходять значення функції. Значення X мають починатися з 1 і збільшуватися на 1 для кожної наступної пари даних. Якщо значення аргументу будуть починатися не з 1 або будуть нецілими, то виникне зупинка "Недопустимий порядковий номер у списку".

Спискова атрибутивна функція типу M відрізняється від спискової числової функції типу L тим, що значення функції можуть бути не лише типу Integer, Real або Name, а й СЧА або виразом у дужках. Аргумент є також цілого типу, його розглядають як порядковий номер, тому він має починатися з 1 і зростати на 1 для кожної наступної пари даних. Наприклад:

```
Oplata FUNCTION X$Kvart,M5  
1, Q$Drog/ 2, N$Term2/ 3, S$Uta/ 4, V$Zvan/ 5, FN$Ost
```

Обчислюють функцію типу M так само, як функцію типу L, лише після ідентифікації аргументу – порядкового номера відбувається обчислення відповідного СЧА або виразу в дужках. Наприклад, при X=4 підраховується арифметична змінна з іменем Zvan. Якщо у функції типу M значення Y є типу Integer, Real або Name, то вона аналогічна функції типу L.

3.1.4. 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 завершення* 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. У моделі може бути багато блоків TERMINATE, але лічильник завершень лише один.

Коли користувач готує модель, він задає час моделювання, зазначаючи в команді START значення лічильника завершень. Оскільки траєкторії руху транзактів у моделі мають різний сенс, то кожен блок TERMINATE може або зменшувати, або не зменшувати значення лічильника завершень.

Розглянемо приклад, коли блок TERMINATE і команду START використовують для керування часом моделювання. Припустимо, що розробник, вибравши за одиницю часу 1 хв, хоче промоделювати поведінку системи протягом восьми годин, після чого моделювання має завершитися. За одиницю модельного часу візьмемо 1 хв, тоді час моделювання дорівнюватиме $8 \cdot 60 = 480$ одиниць.

Будь-яка модель в GPSS складається з одного або кількох сегментів – частин блок-схеми моделі, відокремлених парою блоків GENERATE-TERMINATE. Для керування часом моделювання розробник виконує такі дії:

- вводить до моделі сегмент з блоків GENERATE та TERMINATE;
- у всіх інших блоках TERMINATE моделі використовує операнд A за замовчуванням;
- у команді START на полі операнда A використовує 1.

У процесі моделювання транзакти, які рухаються в інших сегментах моделі, час від часу виводяться з моделі в інших блоках TERMINATE, але вони не змінюють значення лічильника завершень. Наприклад, у моделі з двох сегментів

```
GENERATE 100,40
TERMINATE
GENERATE 480
TERMINATE 1
START 1
```

перший (основний) сегмент моделює процес створення випадкового потоку транзактів, які надходять до моделі з інтервалами модельного часу, рівномірно розподіленими на проміжку [60; 140], і знищення цих танзактів. Оскільки поле операнда A блока TERMINATE у першому сегменті порожнє, то знищені в цьому блоці транзакти не зменшують значення лічильника завершень. У другому сегменті блок GENERATE згенерує перший транзакт у момент модельного часу, який дорівнює 480. Проте цей транзакт виявиться останнім у даному сегменті, оскільки, увійшовши до блока TERMINATE, він пертворить

на нуль значення лічильника завершень, встановлене оператором START, яке дорівнювало 1, тому планувальник припинить моделювання.

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

```
START A,[B],C,[D]
```

Операнд A задає значення лічильника завершень, яке визначає момент завершення прогону моделі, і може бути лише цілим додатним числом. Операнд B – операнд виведення статистики. Він може бути NP (немає виведення даних) або відсутнім. У випадку задання NP стандартний звіт не виводиться. За замовчуванням виводиться стандартний звіт. Операнд 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. Операнд 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 замовчуванням списки у стандартному звіті не виводяться.

Команду START можна зазначити в кінці програми моделі. Тоді після трансляції моделі, тобто створення об’єкта “процес моделювання”, відразу починається моделювання. Цю команду можна також вводити до програми моделі в інтерактивному режимі.

Деколи 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.

Аналоги транзактів – елементи потоків у реальних системах мають різні характеристики. Розглянемо, як ці характеристики присвоюються транзактам.

3.2. Блоки, які змінюють значення параметрів транзактів

3.2.1. Блок ASSIGN

Кожен транзакт може мати будь-яку кількість параметрів з довільною інтерпретацією сенсу кожного з них. У момент генерування транзакта всі його параметри (які використовують в моделі) мають нульові значення. Для присвоювання параметрам початкових значень або зміни цих значень служить блок ASSIGN (присвоювати), який має такий формат:

ASSIGN A,B,[C]

Операнд A задає номер параметра, якому присвоюється значення. Він може бути іменем, додатним цілим числом, виразом у дужках, СЧА, СЧА*параметром і знаками після них + (якщо треба збільшити) або – (якщо треба зменшити).

Операнд B визначає значення, яке потрібно додати, відняти або яким треба замінити значення в параметрі, заданому операндом A. Якщо такого параметра немає, то він автоматично створюється зі значенням, яке дорівнює 0. Операнд B може бути таким самим, як і операнд A, крім того, числом і рядком.

Операнд C задає номер модифікатора-функції. У випадку використання операнда C значення операнда B множиться на значення модифікатора-функції. Отриманий результат множення стає значенням, яке змінює значення параметра, заданого операндом A. Треба пам'ятати, що операнд C визначає номер функції або її ім'я, тому не потрібно використовувати СЧА FN або FN\$ перед ним. Наведемо приклади запису блока ASSIGN:

```
ASSIGN 1,53.7
ASSIGN 2+,Q3
ASSIGN 3-,73.5,2
ASSIGN Name,"Klon"
ASSIGN Chas1-,(Exponential(7,0,Ser)),Nordis
```

У першому прикладі параметру 1 присвоюється число 53,7. У другому прикладі до значення параметра 2 додається значення поточної довжини черги номер 3. У третьому прикладі від значення параметра 3 віднімається результат множення числа 73,5 на підраховане значення функції номер 2. У четвертому прикладі параметру з іменем Name присвоюється рядок Klon. У п'ятому прикладі обчислюються вираз у дужках і функція з іменем Nordis, і результат їхнього добутку віднімається від значення параметра з іменем Chas1.

3.2.2. Блок PLUS

В GPSS World значення параметрів транзактів можна також змінювати за допомогою блока PLUS. Блок PLUS підраховує вираз і може записати його в параметр. Блок PLUS має такий формат:

PLUS A,B

Операнд А – вираз, який може бути іменем, числом, рядком, виразом у дужках, СЧА або СЧА*параметром.

Операнд В – номер параметра транзакта, в якому зберігається результат. Може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Наприклад:

PLUS (Exponential(5,0,15)+71.25),Pidsumok

Після потрапляння транзакта до блока PLUS підраховується вираз у дужках, який записаний у полі операнда А. Потім отриманий результат присвоюється параметрові з іменем Pidsumok. Якщо такого параметра немає, то він автоматично створюється.

3.2.3. Блок INDEX

Змінити значення параметра транзакта можна також за допомогою блока INDEX такого формату:

INDEX А,В

Операнд А – номер параметра транзакта, який може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Операнд В – числове значення, яке додається до значення параметра, вказаного операндом А. Може бути іменем, числом, виразом у дужках, СЧА або СЧА*параметром.

Отриманий результат заноситься до першого параметра транзакта. Наприклад:

INDEX Dod1,(Normal(3,Met,Ser)+X\$Ost)

Після потрапляння транзакта до блока INDEX підраховується вираз у дужках, записаний у полі операнда В, і результат додається до значення параметра з іменем Dod1. Отриманий результат присвоюється параметрові номер 1 транзакта. Якщо цього параметра не існує, то він автоматично створюється. Якщо немає параметра, зазначеного у полі операнда А, то відбувається зупинка внаслідок помилки „Звертання до неіснуючого параметра”.

3.2.4. Блок MARK

Блок MARK використовують для занесення до активного транзакта або його заданого параметра значення абсолютного модельного часу. Цей блок має такий формат:

MARK [А]

Операнд А – номер параметра транзакта, в який автоматично записується значення абсолютного модельного часу в момент потрапляння цього транзакта до блока MARK. Може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

MARK
MARK Vhid

У першому прикладі операнд А не використовується. За замовчуванням транзактові, який увійшов до цього блока, встановлюється поточний час замість часу надходження транзакта до моделі. Цей поточний час використовують для визначення СЧА М1. СЧА М1 обчислюється як різниця значення абсолютного модельного часу (СЧА АС1) і часу, присвоєного транзактові, який дорівнює або часові надходження транзакта до моделі, або часові останнього проходження транзактом блока MARK без операнда А.

У другому прикладі операнд А використовується. Тому значення абсолютного модельного часу заноситься до параметра з іменем Vhid транзакта, який увійшов до цього блока MARK. Якщо цього параметра немає, то він автоматично створюється.

Блок MARK можна використовувати, якщо необхідно визначити час проходження транзакта через певну ділянку моделі. Для цього використовують СЧА MP. Наприклад, якщо на вході ділянки моделі розташувати блок

MARK Marker

то на виході цієї ділянки СЧА MP\$Marker буде містити різницю між поточним модельним часом і часом, занесеним до параметра Marker блоком MARK.

Зауважимо, що у блоках ASSIGN, PLUS, INDEX і MARK під час запису в поля операндів номера параметра транзакта СЧА Р не використовується.

3.3. Займання та звільнення одноканального пристрою

В GPSS елементами, які потребують обслуговування, є транзакти. Вони рухаються в моделі від блока до блока. Якщо в певний момент активності транзакт має намір зайняти ОКП чи БКП, то для цього він заходить (або намагається увійти) до відповідного блока, який описує цей ОКП або БКП.

Блок, який описує ОКП, повинен володіти такими властивостями:

- якщо ОКП вже використовують, то транзакт не може увійти до блока і мусить чекати у черзі або має бути вилучений з моделі;
- якщо ОКП не використовують, то транзакт може увійти до блока. Унаслідок цього статус ОКП змінюється на „зайнятий”.

Блок, що має такі властивості, це – блок SEIZE (зайняти). Потрапляння транзакта до блока SEIZE моделює займання ОКП.

Після завершення обслуговування потрапляння того самого транзакта до блока RELEASE (звільнити) моделює звільнення ОКП і призводить до зміни стану ОКП з „зайнятий” на „незайнятий”.

Формати блоків:

```
SEIZE  A
RELEASE  A
```

В обидвох блоках операнд А – це ім'я або номер відповідного ОКП, який займає (звільняє) транзакт. Може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. У той час як транзакти перебувають у моделі тимчасово, ОКП, які використовують у моделі, існують постійно протягом всього процесу моделювання.

Якщо транзакт заходить до блока SEIZE, то пристрій, зазначений у полі операнда А, перейшовши до стану „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 ОКП, транзакт може пройти через необмежену кількість блоків. Наприклад:

```
SEIZE      Prystriy
.....
RELEASE   Prystriy
```

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

ОКП, 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.

Кожен ОКП має, зокрема, такі СЧА: F – стан пристрою (0 – вільний, 1 – зайнятий); FR – коефіцієнт використання в частках числа 1000; FC – кількість займань пристрою; FT – середній час зайнятості пристрою.

3.4. Імітування обслуговування за допомогою затримки в часі

Зазвичай, транзакт займає ОКП 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 до блоку RELEASE і звільнити ОКП.

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

ADVANCE A,[B]

Тут А – середній час обслуговування, а В – спосіб модифікації операнда А, тобто операнди А і В мають такий самий сенс, як і відповідні операнди блока GENERATE. Кожен з операндів А і В 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 заданий лише операнд А, то блок його підраховує і використовує як час затримки. Наприклад:

ADVANCE (Normal(23,X\$Seredn,X\$SerVidh))

Тут час затримки розподілений за нормальним законом із середнім значенням і середнім квадратичним відхиленням, попередньо записаними командою INITIAL в комірки зберігання з іменами Seredn, і SerVidh відповідно. Для генератора нормально розподіленої випадкової величини джерелом випадкових чисел, рівномірно розподілених на проміжку [0,1], є генератор номер 23 (RN23).

Якщо задані операнди А і В, і В не визначає функцію, то обидва операнди підраховуються (якщо вони не константи) і за час затримки береться значення випадкової величини, рівномірно розподіленої на проміжку (А–В, А+В). Для розігрування випадкових чисел можна вибрати будь-який генератор рівномірно розподілених випадкових чисел. Роблять це так само, як і під час вибору генератора для блока GENERATE. Відмінність полягає лише в тому, що номер генератора на сторінці Random Numbers у журналі Settings моделі потрібно зазначити в полі блока ADVANCE. За замовчуванням використовують генератор випадкових чисел номер 1. Наприклад:

ADVANCE 46.8,23.4

У цьому випадку транзакт затримується у блоці ADVANCE на час, рівномірно розподілений у проміжку [23,4; 70,2].

Так само, як і у блоці GENERATE, незважаючи на спосіб обчислення часу затримки, значення операнда В не повинне перевищувати значення операнда А. Інакше в процесі моделювання відбудеться зупинка внаслідок помилки „Від’ємний приріст часу”.

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 використовує як час затримки. Наприклад:

```
ADVANCE King, FN$Port
```

Тут час затримки – це добуток значень функції з іменем Port і змінної користувача King, якій попередньо має бути присвоєне числове значення командою EQU.

Як було зазначено раніше, у блоці GENERATE можна використовувати функції та арифметичні змінні, попередньо визначені командами FUNCTION і VARIABLE або FVARIABLE. Проте в цих командах заборонені посилання на параметри транзактів, оскільки транзактів ще немає. В операндах блока ADVANCE посилання на параметри транзактів можливі. Цим парараметрам користувач попередньо має присвоїти відповідні значення. Наприклад:

```
ADVANCE P1  
ADVANCE (Exponential(5,0, MX$Ser(P2, P$Stov)))  
ADVANCE SerIntNadh, FN*3
```

У першому прикладі транзакт затримується на час, який дорівнює значенню параметра 1. У другому час затримки визначено за показниковим законом, для якого середнє значення вибирається з елемента матриці, номери рядка і стовпця якого містяться в параметрі 2 і в параметрі з іменем Stov відповідно. Для генератора показникового розподілу джерелом рівномірно розподілених випадкових чисел на проміжку [0, 1] є генератор номер 5 (RN5).

У третьому прикладі час затримки знайдено як добуток значення змінної користувача SerIntNadh і підрахованого значення функції, номер якої міститься в параметрі 3 активного транзакта (тут використовується СЧА з непрямою адресацією).

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 нулеві. У такому випадку транзакт у блоці ADVANCE не затримується і переходить до наступного блока.

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

Використаємо блоки GENERATE, SEIZE, ADVANCE, RELEASE і TERMINATE для моделювання *одноканальної СМО з очікуванням*:

```
GENERATE (Exponential(3,0,10))  
SEIZE System  
ADVANCE (Exponential(5,0,8))  
RELEASE System
```

TERMINATE 1

Тут блок GENERATE за допомогою генератора випадкових чисел номер 3 (RN3) створює транзакти через проміжки часу, розподілені за показниковим законом з середнім значенням 10. Транзакт, увійшовши до блока SEIZE, займе ОКП із символічним іменем System, затримається там завдяки блокові ADVANCE на час, розподілений за показниковим законом з середнім значенням 8, і потім покине цей блок. Після того, як транзакт увійде до блока RELEASE, наступний транзакт намагатиметься використати ОКП System. Транзакти, які виходитимуть з блока GENERATE у моменти зайнятості пристрою, не зможуть увійти до блока SEIZE і будуть залишатися у блоці GENERATE, утворюючи чергу в списку поточних подій.

3.5. Займання та звільнення багатоканального пристрою

Два та більше пристроїв, призначених для обслуговування, можуть бути змодельовані на GPSS відповідною кількістю ОКП, розташованих поруч, тобто паралельно. Це стосується випадку, коли окремі пристрої є різними, тобто мають різні характеристики, наприклад, інтенсивність обслуговування.

Однак часто паралельно працюючі пристрої є однорідними. GPSS передбачає для моделювання однорідних паралельних пристроїв спеціальний засіб – багатоканальний пристрій. БКП може використовуватися кількома транзактами одночасно. Обмежень на кількість БКП в моделі немає. Для того щоб їх розрізнити їм дають різні імена. БКП визначається до його використання командою STORAGE. Її формат такий:

Name STORAGE A

Name – ім'я БКП. Символічному імені можна поставити у відповідність номер командою EQU. Це необхідно зробити, якщо потрібно звертатися до кількох БКП у блоках SELECT і COUNT. Операнд A – кількість каналів обслуговування. Він може бути лише цілим додатним числом.

У моделі можна організують функціонування БКП у двох режимах:

- займання та звільнення БКП;
- недоступність БКП.

Займання та звільнення БКП імітують блоки ENTER (увійти) і LEAVE (вийти, залишити). Формати блоків:

ENTER A,[B]
LEAVE A,[B]

Операнд A в обидвох блоках використовують для позначення імені відповідного БКП. Операнд B задає кількість пристроїв (каналів), які мають бути зайняті у блоці ENTER або звільнені у блоці LEAVE у момент потрапляння до них транзакта. Зазвичай, поле операнда B порожнє, і в цьому випадку за замовчуванням займається або звільняється один канал. Якщо B=0, то блок вважається недієздатним.

Під час входження транзакта до блока ENTER операнд A слугує для знаходження БКП із зазначеним іменем. Якщо такого БКП немає, то відбувається зупинка внаслідок помилки „Звертання до неіснуючої пам'яті”. Якщо БКП існує і задано операнд B, то планувальник його підраховує, округлює до цілого, й отриманий результат використовує для оцінювання вільної ємності (кількості вільних каналів). Транзакт може увійти до блока ENTER, якщо БКП перебуває в доступному стані і достатньо ємності для виконання замовлення. В іншому випадку транзакт переводиться до списку затримки пристрою згідно з його пріоритетом.

Коли транзакт заходить до блока ENTER, планувальник виконує такі дії:

- збільшує на 1 лічильник входжень БКП;
- збільшує на значення операнда B (за замовчуванням на 1) поточний вміст

БКП;

- зменшує на значення операнда В (за замовчуванням на 1) доступну ємність БКП.

Якщо транзакт перед входженням до блока ENTER подає запит на більшу кількість каналів (елементів пам'яті), ніж та, яка визначена командою STORAGE, тобто її операнд А менший від операнда В блока ENTER, то виникає помилка „Запит на елементи пам'яті перевищує її загальну ємність”.

БКП можна перевизначити, тобто змінити ємність ще одною командою STORAGE з тим самим іменем. Наприклад:

```
Vkp STORAGE 3
```

Повторний опис:

```
Vkp STORAGE 5
```

Імітацію обслуговування протягом певного проміжку часу в БКП також здійснюють блоком ADVANCE. Наприклад:

```
Prystr STORAGE 4
```

```
.....
ENTER      Prystr,2
ADVANCE    10,5
LEAVE      Prystr,2
.....
```

Тут команда STORAGE визначає БКП з іменем Prystr ємністю 4 одиниці. Потрапляючи до блока ENTER, транзакт займає 2 канали і стільки ж звільнює, виходячи з блока LEAVE.

У випадку задання фрагмента програми в такому вигляді

```
Prystr STORAGE 4
```

```
.....
ENTER      Prystr,5
ADVANCE    10,5
LEAVE      Prystr,2
.....
```

перед блоком ENTER відбудеться зупинка внаслідок помилки, оскільки транзакт намагається зайняти більше каналів (5), ніж визначено (4) командою STORAGE. Зупинка також станеться, якщо під час виходу з блока LEAVE транзакт намагатиметься звільнити більше каналів, ніж визначено командою STORAGE.

Скористаємося блоками ENTER–LEAVE і командою STORAGE для моделювання двоканальної СМО з очікуванням:

```
System STORAGE 2
GENERATE (Uniform(3,2,10))
ENTER System
ADVANCE (Uniform(5,1,8))
LEAVE System
TERMINATE 1
```

У цьому прикладі операнд В у блоках ENTER і LEAVE не використовується, тому транзакт займає і звільнює одну одиницю ємності БКП. Якщо поточний вміст БКП з іменем System менше ніж 2, тобто у блоці ADVANCE перебуває або один, або немає жодного транзакта, то черговий транзакт, який виходить з блока GENERATE, може увійти до блока ENTER, а потім і до блока ADVANCE. Якщо ж поточний вміст БКП дорівнює 2, то черговий транзакт залишається у блоці GENERATE, утворюючи чергу в списку поточних подій. Після завершення часу затримки одного з двох транзактів, які перебувають у блоці ADVANCE, і після його потрапляння до блока LEAVE перший із заблокованих транзактів зможе увійти до блока ENTER. Зауважимо, що у блоках GENERATE і ADVANCE даної моделі в полі операнда А використано бібліотечний генератор рівномірного розподілу, заданий оператором Uniform(A,B,C). Отже, інтервал надходження транзактів і час обслуговування є випадковими величинами, рівномірно розподіленими на проміжках [2,10] і [1,8] відповідно.

3.6. Перевірка стану одноканальних і багатоканальних пристроїв

3.6.1. Блок GATE

У GPSS World блок GATE (впустити) використовують для визначення стану пристроїв без зміни цього стану. Формат запису:

GATE X A,[B]

Блок GATE працює у двох режимах:

- відмови входження до блока;
- дозволу входження та альтернативного виходу.

Працюючи в режимі відмови, блок GATE не пропускає транзакти, якщо відповідний об'єкт не перебуває в потрібному стані. Якщо ж висунута у блоці GATE умова виконується, то активний транзакт заходить до нього і потім переходить до наступного за порядком блока. Операнд А визначає ім'я або номер апаратного об'єкта.

Умову задає один з таких операторів X, зв'язаних з ОКП або БКП:

- NU – ОКП, заданий операндом А, вільний;
- U – ОКП, заданий операндом А, зайнятий;
- SE – БКП, заданий операндом А, порожній;
- SF – БКП, заданий операндом А, заповнений;
- SNE – БКП, заданий операндом А, непорожній;
- SNF – БКП, заданий операндом А, незаповнений;
- SNV – БКП, заданий операндом А, недоступний;
- SV – БКП, заданий операндом А, доступний.

Отже, тип об'єкта, який перевіряється, неявно задає умовний оператор X.

Операнд В містить номер або ім'я наступного блока для транзакта, який заходить до блока GATE, у випадку, коли умовний оператор не має значення „істина”. Операнди А і В можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Якщо операнд В не використовують, то перевірка здійснюється в режимі відмови. Якщо результат цієї перевірки не буде „істина”, то транзакт переводиться до *списку повторних спроб* тестованого об'єкта. Коли стан об'єкта змінюється, то заблокований транзакт знову активізується і повторюється перевірка заданої блоком GATE умови. Якщо ця умова виконується, то транзактові дозволяється увійти до блока GATE і далі перейти до наступного за порядком блока.

Наприклад:

```
GATE NU Prystriy
GATE NU System,Node
GATE SNF Prystriy2,Mitka
```

У першому прикладі блок GATE не пропустить транзакта за умови зайнятості

ОКП з іменем Prystriy. У другому прикладі у випадку зайнятості ОКП з іменем System транзакт скеровується до блока з міткою Node.

У третьому прикладі, якщо БКП з іменем Prystriy2 незаповнений, тобто є вільні канали, то задана у блоці GATE умова виконується і транзакт буде скерований до наступного блока. Якщо БКП буде заповнений, то транзакт скеровується до блока з міткою Mitka.

Приклад 3.1. Застосування блока GATE для моделювання двоканальної СМО з відмовами.

```

;   Приклад 3.1.
;   Визначення арифметичної змінної та БКП
Imo  FVARIABLE  N$Mitka/N$Vsi
Sto  STORAGE 2
;   Сегмент імітування надходження та розподілу замовлень
Vsi  GENERATE (Exponential(1,0,5))
      GATE SNF Sto,OUT
;   Імітування роботи БКП
      ENTER Sto
      ADVANCE (Uniform(2,5,15))
      LEAVE Sto
Mitka TERMINATE
;   Імітування відмов (знищення транзактів)
OUT  TERMINATE
;   Сегмент задання часу моделювання та збереження значення
;   арифметичної змінної
      GENERATE 1000
      SAVEVALUE Imobs,V$Imo
      TERMINATE 1
      START 1

```

Якщо БКП з іменем Sto незаповнений, тобто є вільні канали, то задана у блоці GATE умова виконується і транзакт буде скерований до блока ENTER. Якщо БКП буде заповнений, то транзакт скеровується до блока з міткою OUT і, не проходячи через блок TERMINATE з міткою Mitka, вилучається з моделі. Так моделюється втрата замовлення для даної СМО з відмовами. Всі транзакти, які здійснили обслуговування в БКП Sto, проходячи через блок TERMINATE з міткою Mitka, поповнюють лічильник входжень цього блока. Значення лічильника N\$Mitka фігурує в арифметичній змінній з іменем Imo, визначеній на початку моделі, і яку використовують для обчислення імовірності обслуговування. У момент, коли модельний час досягає значення 1000, транзакт, створений блоком GENERATE з операндом A=1000, потрапляє до блока SAVEVALUE, завдяки чому обчислюється значення V\$Imo змінної Imo. Отриманий результат призначається комірці зберігання Imobs і виводиться на друк у стандартному звіті моделі. Наголосимо, що в цьому прикладі розглянуто СМО з найпростішим потоком замовлень і рівномірно розподіленим часом

обслуговування.

Для перевірки стану апаратних об'єктів можна використовувати також булеві змінні та блок TEST.

3.6.2. Блок TEST

Блок TEST (перевірити) описує умову, яку перевіряє транзакт, перебуваючи перед входженням до нього, і визначає наступний блок для цього транзакта залежно від того, чи виконується висунута у блоці умова, чи ні. Формат запису блока TEST:

TEST X A,B,[C]

Операнди A і B є порівнюваними величинами. Вони можуть бути іменем, числом, рядком, виразом у дужках, СЧА або СЧА*параметром. Умовний оператор X може бути вираженим одним з шести умовних операторів: L – менше; LE – менше або дорівнює; E – дорівнює; NE – не дорівнює; G – більше; GE – більше або дорівнює.

Якщо операнд C зазначений, то транзакт завжди може увійти до блока TEST і залежно від співвідношення значень операндів A та B буде переданий або до наступного блока, або до блока, вказаного операндом C. Якщо поле операнда C порожнє, то транзакт у випадку невиконання умови не зможе увійти до блока TEST. Він переводиться до *списку повторних спроб* всіх об'єктів, які беруть участь у перевірці умови. Коли стан будь-якого з цих об'єктів змінюється, то транзакт зі списку повторних спроб активізується і перевірка здійснюється знову. Якщо умова виконується, то транзакт отримує дозвіл на вхід до блока TEST.

У булевій змінній допускається використання таких логічних операторів:

- F дорівнює 1, якщо ОКП зайнятий, у протилежному випадку – 0;
 - SF дорівнює 1, якщо БКП заповнений цілковито, інакше – 0;
 - SE дорівнює 1, якщо БКП порожній, інакше – 0;
 - SV дорівнює 1, якщо БКП доступний, інакше – 0,
- а також СЧА ОКП та БКП:
- FC – кількість транзактів, які займали ОКП за допомогою блоків SEIZE і PREEMPT;
 - FR – коефіцієнт використання ОКП;
 - FT – середній час обслуговування ОКП одного транзакта;
 - S – кількість зайнятих каналів БКП;
 - SA – середнє значення кількості зайнятих каналів БКП;
 - SC – лічильник використання БКП;
 - SR – коефіцієнт використання БКП;
 - ST – середній час використання одного каналу БКП.

Отже, СЧА SE, SF і SV можна використовувати як у блоці GATE, так і в булевих змінних блока TEST. Якщо ж у блоці GATE у полі умовного оператора

Х записати, наприклад SA, то під час трансляції виникне помилка „Недопустиме використання специфікатора класу СЧА”. Якщо ж у булевій змінній блока TEST записати, наприклад SNV, то також буде помилка „Припускається фактор виразу”.

Наведемо приклад використання булевої змінної та блока TEST для перевірки стану ОКП.

```
Perev BVARIABLE   F$Server
.....
TEST E           BV$Perev,0,Mitka
SEIZE            Server
ADVANCE          P$Obrob
RELEASE         Server
```

У цьому прикладі під час потрапляння транзакта до блока TEST обчислюється булева змінна Perev. У ній використано СЧА ОКП F (тут використання умовних операторів NU та U не допускається). Якщо ОКП з іменем Server зайнятий, то значення змінної Perev дорівнює 1. Тоді умова, задана блоком TEST, не виконується, оскільки $1 \neq 0$, і транзакт не може зайняти ОКП, а переходить до блока з міткою Mitka, вказаною операндом C. Якщо ж ОКП з іменем Server вільний, то булева змінна Perev дорівнює 0. Тоді умова, яку перевіряє блок TEST, виконується, і транзакт переходить до наступного блока, тобто займає ОКП з іменем Server.

Блоки GATE, TEST, а також блок TRANSFER, які належать до групи блоків, які змінюють маршрути руху транзактів, використовують для моделювання різних СМО з втратами замовлень. Приклад використання блока GATE для моделювання СМО з відмовами наведений вище. Інші приклади таких моделей ми розглянемо після ознайомлення з блоком TRANSFER.

3.7. Методи зміни маршрутів руху транзактів

Для зміни маршрутів руху транзактів у моделі, крім блоків GATE і TEST, застосовуються також блоки TRANSFER, DISPLACE та LOOP.

3.7.1. Блок TRANSFER

Блок TRANSFER (передати) слугує для скерування транзакта, який потрапив до цього блока, до будь-якого іншого блока моделі. Блок TRANSFER має такий формат:

TRANSFER [A],[B],[C],[D]

Усі режими блока TRANSFER, крім безумовного, *вибіркові*, тобто відрізняються один від одного способом вибору чергового блока, до якого має бути скерований активний транзакт. Операнд А задає цей режим вибору. Блок TRANSFER може працювати в дев'яти різних режимах:

- , (за замовчуванням) – безумовний;
- . – статистичний, тобто вибір випадковим чином одного з двох блоків;
- BOTH – послідовний вибір одного з двох блоків;
- ALL – послідовний вибір одного з кількох блоків;
- PICK – вибір випадковим чином одного з кількох блоків;
- FN – функціональний;
- P – параметричний;
- SBR – підпрограмний;
- SIM – одночасний.

Операнд А, крім перерахованих значень, також може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Операнди В і С визначають можливі імена наступних блоків або їхнє розташування. Вони також можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Використання цих значень ми опишемо далі під час ознайомлення з конкретними режимами роботи. Якщо операнд В відсутній, то планувальник записує замість нього ім'я або номер блока, який розташований після блока TRANSFER.

Розглянемо режими роботи блока TRANSFER.

3.7.1.1. Режим безумовної передачі

У режимі безумовної передачі операнд А не використовують, а операнд В зазначає ім'я блока, до якого скеровується транзакт. Блок TRANSFER завжди відчинений для входження транзактів. Наприклад:

TRANSFER ,Mit1

Потрапивши до блока TRANSFER, транзакт відразу намагається увійти до блока з міткою Mit1. Якщо він отримує відмову на входження, то залишається у

блоці TRANSFER.

3.7.1.2. Режим статистичної передачі

Коли операнд А використовується і не є ключовим словом, то блок TRANSFER працює в режимі передачі транзакта до одного із двох блоків випадковим чином.

Значення операнда А визначає імовірність, з якою транзакт скеровується до блока, зазначеного в полі операнда С. З імовірністю $1-A$ транзакт скеровується до блока, вказаного в полі операнда В (до наступного блока, якщо поле В порожнє).

Імовірність у полі А може бути задана безпосередньо десятковим дробом, який починається з крапки. Наприклад, блок

TRANSFER .75,This,That

з імовірністю 0,75 скеровує транзакти до блока з іменем That, а з імовірністю 0,25 – до блока з іменем This.

Якщо ж поле А починається не з десяткового дробу і не містить одного з ключових слів – ознак інших режимів роботи, то його значення розглядається як кількість тисячних часток імовірності передачі. Наприклад, попередній блок TRANSFER можна записати також у такому вигляді:

TRANSFER 750,This,That

Числове значення операнда А можна задавати будь-яким СЧА. Можливі такі випадки:

- значення операнда А менше або дорівнює нулю;
- значення операнда А дорівнює або більше ніж 1000;
- значення операнда А більше від нуля, але менше ніж 1000.

Якщо підраховане значення операнда А менше або дорівнює нулю, то буде здійснюватися безумовне скерування транзакта до блока В. Якщо значення операнда А більше або дорівнює 1000, то буде здійснюватися безумовне скерування транзакта до блока С. У третьому випадку блок TRANSFER працює в звичайному режимі.

Наприклад, запис

TRANSFER .P3,,That

означає, що тризначне число, записане в параметрі 3 транзакта, який потрапив до блока TRANSFER, інтерпретується як імовірність (в частках тисячі) того, що транзакт передаватиметься до блока That, а в решті випадків – до наступного блока, оскільки операнд В не використовується.

Режим статистичної передачі зручно використовувати, наприклад, у таких випадках. Нехай потрібно побудувати імітаційну модель роботи цеху, в якому виробляють деталі, і відомо, що 6,35% виготовлених деталей є бракованими. У моделі це можна реалізувати так:

.....
TRANSFER .635,Kond,Brak

Тоді транзакти, які імітують виготовлені в цеху деталі, в 6,35% випадків скероватимуться до блока з міткою Brak, а в решті 93,65% випадках – до блока з міткою Kond.

Можна визначити генератор – джерело випадкових чисел для блока TRANSFER. Для цього, вибравши в журналі Settings сторінку Random Numbers, треба ввести номер генератора в полі блока TRANSFER.

3.7.1.3. Режим BOTH

Якщо в полі операнда A використовується ключове слово BOTH (обидва), то блок TRANSFER працює в режимі BOTH. У цьому режимі кожний транзакт, увійшовши до блока TRANSFER, перевіряє дві можливості. Спочатку він намагається увійти до блока B (або до наступного блока, якщо поле B порожнє), а якщо це не вдається, тобто блок B не дозволяє входження, то – до блока C. Якщо й ця спроба невдала, то транзакт затримується у блоці TRANSFER до такої зміни умов у моделі, які б дозволили увійти до одного з блоків B або C. Блокові B надається перевага, якщо така можливість виникне одночасно.

Приклад 3.2. Використання блоків TRANSFER для моделювання двоканальної СМО з відмовами і повторними спробами.

```
; Приклад 3.2.  
; Визначення арифметичної змінної та БКП  
Imo FVARIABLE N$Mitka/N$Vsi  
Sto STORAGE 2  
; Сегмент імітування надходження та розподілу замовлень  
Vsi GENERATE (Exponential(1,0,5))  
Ent TRANSFER BOTH,,Refus  
; Імітування роботи БКП  
ENTER Sto  
ADVANCE (Uniform(3,5,25))  
LEAVE Sto  
Mitka TERMINATE  
; Імітування статистичного розподілу замовлень  
Refus TRANSFER .1,,OUT  
ADVANCE (Uniform(2,5,15))  
TRANSFER ,Ent  
OUT TERMINATE  
; Сегмент задання часу моделювання та збереження значення  
арифметичної змінної  
GENERATE 1000  
SAVEVALUE Imobs,V$Imo  
TERMINATE 1  
START 1
```

Транзакти, створені блоком GENERATE, потрапляють до блока TRANSFER з іменем Ent, який працює в режимі BOTH. Якщо в момент надходження транзакта до БКП Sto хоча б один канал вільний, то блок TRANSFER скерує транзакт до наступного блока, тобто до блока ENTER. Якщо ж у момент надходження обидва канали БКП зайняті, і тому блок ENTER не дозволяє входження, то транзакт буде скерований до блока TRANSFER з іменем Refus, який працює в статистичному режимі. З імовірністю 0,9 транзакти з цього блока передаються до наступного блока, затримуються в ньому на випадковий час (рівномірно розподілений на проміжку [5,15]), і за допомогою блока TRANSFER, який працює в безумовному режимі, передаються знову на початок моделі до блока з іменем Ent. З імовірністю 0,1 транзакти з блока з іменем Refus передаються до блока TERMINATE з іменем OUT для знищення.

Зазначимо, що час моделювання встановлюється за допомогою блока GENERATE з операндом $A=1000$, а використання арифметичної змінної Imo і блока SAVEVALUE забезпечує виведення в стандартному звіті ймовірності обслуговування в комірці зберігання Imobs.

Для моделювання тої самої СМО можна використати також блок TEST, замінивши рядок програми

```
Ent TRANSFER BOTH,,Refus
```

на такий:

```
Ent TEST L S$Sto,2,Refus
```

У цьому варіанті моделі транзакт проходить до блока ENTER, якщо поточна кількість зайнятих каналів (СЧА S\$Sto) менша ніж 2.

Можливий ще один варіант цієї моделі з використанням блока GATE. Для його отримання досить у попередній програмі замінити рядок з міткою Ent таким:

```
Ent GATE SNF Sto,Refus
```

Тут транзакт проходить до блока ENTER, якщо умова "БКП Sto незаповнений" є істинною.

3.7.1.4. Режим ALL

Блок TRANSFER у режимі ALL, як і в режимі BOTH, виконує роль „диспетчера”, але на відміну від блока BOTH, він розподіляє транзакти між кількома пристроями.

Для роботи в режимі ALL необхідно в полі операнда A використати ключове слово ALL. У цьому режимі, увійшовши до блока TRANSFER, транзакт перевіряє можливість входження до будь-якого блока, починаючи з блока, зазначеного операндом B, і закінчуючи блоком, вказаним операндом C. Операнд D визначає крок d зміни номера блока, який перевіряється. Значення кроку дає змогу опитувати певні блоки, розташовані між тими, які задані операндами B і C.

Спочатку транзакт, увійшовши до блока TRANSFER, намагається увійти до блока В. Якщо блок В зайнятий, то транзакт намагається увійти до блоків з номерами $v+d$, $v+2d$, ..., $\xi=v+nd$, де v – номер блока В, ξ – номер блока С, n – деяке натуральне число.

Якщо операнд С не використовують, то перевіряється лише один блок.

Оскільки, зазвичай, у полях операндів В і С записують мітки блоків, то блоки потрібно розташовувати так, щоб під час присвоєння номера різниця між номерами блоків, зазначених операндами В і С, була кратною крокові, вказаному в полі операнда D. Наприклад:

TRANSFER ALL,Mit1,Mit2,4

Тут режим ALL допустимий, якщо різниця між номерами, присвоєними блокам Mit1 і Mit2, кратна 4. У моделі це може виглядати так.

Приклад 3.3. *Замовлення розподіляють між трьома БКП. Чергове замовлення обслуговується тим БКП, який першим звільнився.*

```
; Приклад 3.3.
; Визначення трьох БКП
Sto1 STORAGE 2
Sto2 STORAGE 2
Sto3 STORAGE 2
; Сегмент імітування надходження та розподілу замовлень
GENERATE (Exponential(1,0,2))
TRANSFER ALL,Mit1,Mit2,4
; Імітування роботи БКП1
Mit1 ENTER Sto1
ADVANCE (Uniform(3,5,25))
LEAVE Sto1
TERMINATE
; Імітування роботи БКП2
ENTER Sto2
ADVANCE (Uniform(2,5,15))
LEAVE Sto2
TERMINATE
; Імітування роботи БКП3
Mit2 ENTER Sto3
ADVANCE (Uniform(4,3,7))
LEAVE Sto3
TERMINATE
; Сегмент задання часу моделювання
GENERATE 1000
TERMINATE 1
START 1
```

Після надходження транзакта до блока TRANSFER послідовно перевіряється можливість входження спочатку до блока ENTER, позначеного

міткою Mit1. Потім, якщо цей блок не дозволяє входження, до другого блока ENTER без мітки, оскільки різниця між номерами цього блока та блока з міткою Mit1 дорівнює 4. Якщо неможливе входження до другого блока ENTER, то перевіряється третій блок ENTER з міткою Mit2. Якщо й він зайнятий, то транзакт залишається у блоці TRANSFER, і з кожною зміною поточного модельного часу перевірки можливості входження знову повторюються спочатку, тобто з блока ENTER з міткою Mit1.

У блоці TRANSFER операнди B і C можуть бути не мітками, а номерами блоків. Зокрема, якщо в моделі прикладу 3.3 записати

TRANSFER ALL,3,11,4

то результат моделювання буде такий самий (нумерація починається з блока GENERATE).

Зауважимо, що коли планувальник під час перегляду списку поточних подій знаходить транзакт, затриманий у блоках TRANSFER BOTH або TRANSFER ALL, він намагається дозволити просування транзактові, починаючи з блока B. Як наслідок, у режимі BOTH у тих випадках, коли можливий перехід до обидвох блоків B і C, блок B має деяку перевагу. Аналогічно й у режимі ALL у випадку, коли можливий перехід до декількох блоків, то блоки з меншими номерами мають деяку перевагу перед блоками з більшими номерами.

Підтвердимо це результатами моделювання прикладу 3.3:

STORAGE.	ENTRIES	AVE.C.	UTIL.
STO1	123	1.798	0.899
STO2	164	1.595	0.798
STO3	210	1.042	0.521

Результати моделювання показують, що найбільший коефіцієнт використання (UTIL.) має БКП Sto1, а найменший – БКП Sto3, хоч середній час обслуговування, який задається блоком ADVANCE, для БКП Sto3 менший, ніж для БКП Sto1, і він за один і той самий час міг би обслужити більшу кількість транзактів, тобто бути більше завантаженим.

3.7.1.5. Режим PICK

Для роботи блока TRANSFER у режимі PICK необхідно в полі операнда A використати ключове слово PICK. У цьому режимі з послідовності блоків з номерами $v, v+1, v+2, \dots, n$, де v – номер блока, зазначеного операндом B, а n – номер блока, вказаного операндом C, випадково вибирається номер одного блока, до якого має скеровуватися транзакт. Всі блоки, враховуючи й вказані операндами B і C, вибираються з однаковою ймовірністю $1/[(n-v)+1]$. Транзакт намагається перейти лише до вибраного для нього блока. Якщо транзакт не може відразу перейти до наступного блока, то він чекатиме у блоці TRANSFER до того часу, поки не буде знята умова блокування. Номер блока C має бути більший або дорівнювати $v+1$.

Приклад 3.4. *Замовлення розподіляються між чотирма БКП, які мають однаковий час обслуговування. Ймовірність потрапляння до будь-якого з БКП $P=1/4=0,25$. Якщо в режимах BOTH і ALL чергове замовлення займало той БКП, який раніше звільнявся, то в режимі PICK замовлення чекає на звільнення того БКП, на який воно було розподілене. Оскільки блоки, до яких скеровуються замовлення, не розташовані в моделі послідовно, то використовується послідовність блоків TRANSFER у режимі безумовної передачі.*

```

;   Приклад 3.4.
;   Визначення чотирьох БКП
Sto1  STORAGE 3
Sto2  STORAGE 3
Sto3  STORAGE 3
Sto4  STORAGE 3
;   Сегмент імітування надходження та розподілу замовлень
      GENERATE (Exponential(1,0,2))
      TRANSFER  PICK,Mit1,Mit2
Mit1  TRANSFER  ,Mit3
      TRANSFER  ,Mit4
      TRANSFER  ,Mit5
Mit2  TRANSFER  ,Mit6
;   Імітування роботи БКП1
Mit3  ENTER Sto1
      ADVANCE (Uniform(3,5,25))
      LEAVE Sto1
      TERMINATE
;   Імітування роботи БКП2
Mit4  ENTER Sto2
      ADVANCE (Uniform(3,5,25))
      LEAVE Sto2
      TERMINATE
;   Імітування роботи БКП3
Mit5  ENTER Sto3
      ADVANCE (Uniform(3,5,25))
      LEAVE Sto3
      TERMINATE
;   Імітування роботи БКП4
Mit6  ENTER Sto4
      ADVANCE (Uniform(3,5,25))
      LEAVE Sto4
      TERMINATE
;   Сегмент задання часу моделювання
      GENERATE 1000000
      TERMINATE 1

```

START 1

Наведемо фрагмент результатів моделювання прикладу 3.4.

STORAGE	ENTRIES	AVE.C.	UTIL.
STO1	125754	1.881	0.627
STO2	125195	1.877	0.626
STO3	124927	1.876	0.625
STO4	124798	1.874	0.625

Отже, коефіцієнти використання всіх БКП однакові й дорівнюють 0,625-0,627. Незначні розбіжності в кількості входжень до БКП викликані генераторами випадкових чисел.

3.7.1.6. Функціональний режим

У функціональному режимі блок TRANSFER працює тоді, коли замість операнда А використовується ключове слово FN. Під час входження транзакта до блока TRANSFER підраховується значення функції, ім'я якої задане операндом В. Якщо результат не цілий, то від нього береться ціла частина. Для визначення номера наступного блока отримане значення додається до значення операнда С (значення останнього може бути записане нулем). Якщо блок з обчисленим номером зайнятий, то транзакт залишається у блоці TRANSFER до того часу, поки не зможе перейти саме до цього блока. Наприклад:

```
TRANSFER FN,Vybir,P4  
TRANSFER FN,8,P4
```

У першому прикладі транзакт скеровується до блока, номер якого планувальник визначає як суму обчисленого та округленого до цілого значення функції з іменем Vybir і значення параметра 4 цього транзакта. У другому прикладі все так само, лише функція задана не іменем, а номером.

3.7.1.7. Параметричний режим

Для задання параметричного режиму потрібно в полі операнда А вказати ключове слово Р. У цьому режимі активний транзакт скеровується до блока, номер якого визначається як сума значення параметра та значення операнда С. Якщо операнд С не використовують, то значення параметра приймається за номер нового блока. Наприклад:

```
TRANSFER P,Novyi,7  
TRANSFER P,8,7
```

У першому прикладі транзакт скеровується до блока, номер якого дорівнює сумі значення параметра з іменем Novyi і значення операнда С, тобто 7. У другому прикладі все так само, лише параметр заданий не іменем, а номером.

3.7.1.8. Підпрограмний режим

Для задання підпрограмного режиму потрібно в полі операнда А зазначити ключове слово SBR. У цьому режимі активний транзакт завжди скеровується до блока, номер якого вказаний операндом В. Номер блока TRANSFER заноситься до параметра, вказаного операндом С. Наприклад:

TRANSFER SBR,Novyi,Vist

Увійшовши до блока, транзакт скеровується до блока з міткою Novyi, а номер блока TRANSFER записується до параметра з іменем Vist. Якщо такого параметра немає, то планувальник його створює.

Для повернення з підпрограми використовують блок TRANSFER у параметричному режимі.

3.7.1.9. Одночасний режим

Режим SIM (одночасний) використовують у випадку, коли потрібне одночасне виконання кількох умов. Кожен транзакт має свій індикатор затримки SIM. У цьому індикаторі записується результат кожної спроби входження транзакта до наступного блока. Якщо планувальник виявляє умови, які перешкоджають входженню транзакта до блока, то індикатор SIM цього транзакта дорівнює 1. Якщо всі умови переходу до наступного блока задовольняються, то індикатор SIM дорівнює нулеві. Якщо не виконується хоча б одна з умов, то індикатор SIM дорівнює 1. У таких випадках операнд С вказує блок, у якому перевірялась перша умова, і транзакт виконує перевірки всіх умов, поки вони не будуть задовольнятися одночасно. Блок ADVANCE також встановлює індикатор SIM, який дорівнює нулеві. Перевірка станів пов'язана з блоками, які можуть затримувати транзакти. Для перевірки умов використовують блок GATE.

Наприклад:

```
Mit  GATE SE      Sto1
      GATE SE      Sto2
      GATE SE      Sto3
      TRANSFER    SIM,,Mit
      ENTER       Sto4
```

У наведеному прикладі транзакт не може перейти до блока ENTER, якщо БКП Sto1, Sto2 і Sto3 одночасно порожні. Під час входження транзакта до блока TRANSFER перевіряється значення його індикатора SIM. Якщо транзакт був затриманий в якомусь з блоків GATE, то його індикатор дорівнює 1, і під час перевірки індикатора у блоці TRANSFER виявляється, що транзакт був затриманий. Тому він відсилається до того блока, де здійснювалася перша перевірка (Mit), і вся послідовність перевірок повторюється.

Під час затримки транзактів у блоках ASSEMBLE, GATHER або MATCH індикатор SIM не дорівнює 1.

Зауважимо, що режим SIM використовують зрідка, оскільки за допомогою

булевих змінних і блока TEST можна ефективніше керувати станом великої кількості об'єктів.

3.7.2. Блок LOOP

Блок LOOP призначений для організації в моделі циклів. Він має такий формат:

LOOP A,B

Операнд А – параметр транзакта або параметр циклу, в якому міститься число – кількість повторень якого-небудь сегмента моделі. Операнд В – мітка блока, з якого починається цикл. Обидва операнди можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

LOOP KilPovt,Pochat

Нехай блок з міткою Pochat є початком циклу, тобто розташований раніше від блока LOOP. Коли транзакт, пройшовши ділянку моделі, яка починається блоком з міткою Pochat, увійде до блока LOOP, значення його параметра з іменем KilPovt зменшується на 1. Якщо це значення не дорівнює нулеві, то транзакт переходить до блока з міткою Pochat, тобто цикл повторюється. Якщо ж після віднімання 1 значення параметра циклу дорівнює нулю, отже, виконана задана кількість повторень, то транзакт переходить до наступного блока.

Якщо під час входження до блока LOOP виявиться, що транзакта з іменем KilPovt немає, то станеться зупинка внаслідок помилки „Звертання до неіснуючого параметра”.

Внесемо деякі зміни до моделі прикладу 3.2.

Приклад 3.5. Використання блока LOOP для моделювання двоканальної СМО з відмовами і заданою кількістю повторних спроб.

```
; Приклад 3.5.  
; Визначення арифметичної змінної та БКП  
Imo FVARIABLE N$Mitka/N$Vsi  
Sto STORAGE 2  
; Сегмент імітування надходження й розподілу замовлень  
; та присвоєння параметрові транзакта кількості повторень  
Vsi GENERATE (Exponential(1,0,5))  
ASSIGN 1,3  
Ent TRANSFER BOTH,,Refus  
; Імітування роботи БКП  
ENTER Sto  
ADVANCE (Uniform(3,5,25))  
LEAVE Sto  
Mitka TERMINATE  
; Імітування статистичного розподілу замовлень  
Refus ADVANCE (Uniform(2,5,15))
```

```

LOOP 1,Ent
TERMINATE
; Сегмент задання часу моделювання та збереження значення ;
арифметичної змінної
GENERATE 1000
SAVEVALUE Imobs,V$Imo
TERMINATE 1
START 1

```

Транзакт, створений блоком GENERATE, надходить до блока ASSIGN, де його першому параметру присвоюється значення 3 (кількість повторних спроб). Після цього, як і в моделі прикладу 3.2, транзакти потрапляють до блока TRANSFER з іменем Ent, який працює в режимі BOTH. Якщо в момент надходження транзакта в БКП Sto хоча б один канал вільний, то блок TRANSFER скерує транзакт до наступного блока, тобто до блока ENTER. Якщо ж у момент надходження обидва канали БКП зайняті, і тому блок ENTER не дозволяє входження, то транзакт буде скерований до блока ADVANCE з іменем Refus і затримається в ньому на випадковий час (рівномірно розподілений на проміжку [5,15]). Після очікування транзакт заходить до блока LOOP, де значення його параметра номер 1 зменшується на 1. Якщо це значення не дорівнює нулеві, то транзакт переходить до блока з міткою Ent (вказаному як початок циклу у полі операнда В блока LOOP) і робить чергову спробу ввійти до блока ENTER, тобто цикл повторюється. Якщо ж після віднімання 1 значення параметра циклу дорівнює нулю, отже, виконана задана кількість повторень (яка дорівнює 3), то транзакт переходить до наступного блока TERMINATE для знищення.

У моделі також передбачено обчислення ймовірності обслуговування ($P_{обс}$). Нижче наведені значення $P_{обс}$, отримані після кількох прогонів цієї моделі з різними значеннями кількості повторних спроб потрапити на обслуговування.

Кількість спроб	$P_{обс}$	Кількість спроб	$P_{обс}$
1	0,477	8	0,641
2	0,549	10	0,651
3	0,569	20	0,662
4	0,595	30	0,667
5	0,595	100	0,667

Дані таблиці показують, що зі зростанням кількості спроб імовірність обслуговування спочатку зростає, але потім, досягнувши максимального значення 0,667, $P_{обс}$ стабілізується, вичерпавши максимальні можливості даної СМО.

3.7.3. Блок DISPLACE

Блок DISPLACE призначений для відшукування будь-якого транзакта і переміщення його до нового блока. Блок має такий формат:

DISPLACE A,B,[C],[D]

Операнд А – номер транзакта, який потрібно перемістити; операнд В – мітка блока, до якого передається транзакт, вказаний операндом А. Операнд С – номер параметра цього транзакта, до якого записується час, який залишився до завершення його обслуговування, якщо транзакт перебував у списку майбутніх подій. Операнд D – мітка альтернативного блока для активного транзакта.

Всі операнди можуть бути іменем додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

DISPLACE (P3+24),Mod,Zal,Mit

Тут операнд А є виразом у дужках. Цей вираз підраховується та округлюється до цілого. Отриманий результат є номером транзакта, який треба перемістити. Далі блок DISPLACE розшукує цей транзакт. Для останнього можливі випадки:

- транзакт є в моделі і не перебуває у списку майбутніх подій;
- транзакт є в моделі і перебуває у списку майбутніх подій;
- транзакта із зазначеним номером немає в моделі.

У першому випадку транзакт передається до блока з іменем Mod. У другому випадку визначається час, який залишився до його повернення до процесу моделювання, і записується до параметра з іменем Zal. Якщо параметра з таким іменем немає, то він автоматично створюється. Транзакт також передається до блока з міткою Mod. У третьому випадку, тобто коли в моделі немає транзакта з потрібним номером, активний транзакт, який увійшов до блока DISPLACE, скеровується до блока з міткою Mit. Якщо операнда D немає, то активний транзакт переходить до наступного блока.

Коли транзакт переходить до нового блока, він вилучається зі списків:

- майбутніх подій;
- відкладених переривань (для транзактів, які здійснюють переривання);
- затримки (у порядку пріоритету);
- користувача;
- повторних спроб.

Одночасно він не вилучається зі списків:

- поточних подій;
- переривань (для перерваних транзактів);
- груп.

Під час пересування перервані виконання у пристроях не припиняються. Це означає, що транзакт продовжує займати пристрій.

3.8. Методи реєстрації статистичних даних

3.8.1. Об'єкти типу „черга”

У GPSS об'єкти типу “черга” вводяться для реєстрації статистичних даних. Ця статистика повинна містити таку інформацію:

- кількість транзактів, які стали у чергу;
- кількість транзактів, які фактично приєднались до черги, і кількість таких, що потрапили на обслуговування без черги;
- максимальне значення довжини черги;
- середня довжина черги;
- середній час очікування в черзі.

GPSS забезпечує можливість отримання такої статистики за допомогою так званого *реєстратора черги*. У випадку використання реєстратора черги в тих точках моделі, де ресурси обмежені, планувальник автоматично збирає статистику, яка описує очікування (якщо воно є) в цих точках.

Реєстратори черг розрізняють, присвоюючи їм імена. Розробник вносить реєстратор черги до моделі за допомогою пари взаємодоповнюючих блоків QUEUE (стати в чергу) і DEPART (покинути чергу).

Формат запису блока QUEUE такий:

```
QUEUE A,[B]
```

Блок QUEUE збільшує довжину черги. Операнд А задає номер або ім'я черги, а операнд В – кількість одиниць, на яку збільшується поточна довжина черги під час входження транзакта до блока QUEUE. Якщо операнд В не використовують, то додається одиниця.

Розглянемо приклади запису блока QUEUE:

```
QUEUE Cherha
```

```
QUEUE P3,P2
```

У першому прикладі блок QUEUE збільшує на одиницю довжину черги Cherha після входження кожного транзакта. У другому прикладі довжина черги, номер якої заданий у параметрі номер 3 транзакта, збільшується на кількість одиниць, задану в параметрі 2.

Формат запису блока DEPART має вигляд:

```
DEPART A,[B]
```

Блок DEPART слугує для зменшення довжини черги. Операнд А задає номер або ім'я черги, довжину якої треба зменшити. Операнд В задає кількість одиниць, на яку зменшується довжина черги. Якщо операнд В не використовується, то за замовчуванням довжина черги зменшується на 1.

Операнди А і В у блоках QUEUE і DEPART можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

```
DEPART Cherha
```

DEPART V2,(V3+5.4)

У першому прикладі блок DEPART зменшує довжину черги Serha на одиницю. У другому прикладі операнди A і B задані арифметичною змінною V2 і виразом у дужках, який також містить арифметичну змінну V3. Під час входження транзакта до блока DEPART змінна й вираз у дужках підраховуються та округлюються. Після цього довжина черги, номер якої заданий значенням змінної V2, зменшується на округлене значення виразу в дужках.

Черзі дозволено записувати в полі операнда A номер без попереднього присвоєння його імені командою EQU. Наприклад:

```
QUEUE 4
.....
DEPART 4
```

Для отримання статистики про транзакти, заблоковані перед яким-небудь блоком моделі, блоки QUEUE і DEPART розташовують перед і після цього блока відповідно. Під час проходження транзакта через блоки QUEUE і DEPART відповідним чином змінюються такі СЧА черг: Q – поточна довжина черги; QM – максимальна довжина черги; QA – середня довжина черги; QC – загальна кількість транзактів, які стали в чергу; QZ – кількість транзактів, які пройшли через чергу без очікування (кількість „нульових” входжень); QT – середній час перебування транзакта в черзі з урахуванням „нульових” входжень; QX – середній час очікування без врахування „нульових” входжень.

Приклад 3.6. Використання блоків QUEUE і DEPART для отримання статистики про чергу у випадку моделювання двоканальної СМО з очікуванням:

```
; Приклад 3.6.
; Задання БКП
System STORAGE 2
; Сегмент імітування надходження та обслуговування
; замовлень і збору статистики очікування
GENERATE (Uniform(3,2,10))
QUEUE 1
ENTER System
DEPART 1
ADVANCE (Uniform(5,2,20))
LEAVE System
TERMINATE
; Сегмент задання часу моделювання та збереження СЧА QA
GENERATE 1000
SAVEVALUE SerDov,QA1
TERMINATE 1
START 1
```

Очікування може виникнути через зайнятість БКП з іменем System. Для

отримання статистики про очікування введено реєстратор черги з номером 1. Транзакти, заблоковані перед блоком ENTER через зайнятість БКП System, потрапляють до блока QUEUE і вносять свій вклад у статистику черги. Після звільнення пристрою System перший із заблокованих транзактів увійде до блока ENTER і одночасно до блока DEPART, припиняючи накопичення статистики про очікування цього транзакта.

Наведемо фрагмент стандартного звіту, отриманого після реалізації моделі прикладу 3.6.

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)
1	6	0	169	45
	AVE.CONT.	AVE.TIME	AVE.(-0)	
	0.983	5.819	7.931	
SAVEVALUE	RETRY	VALUE		
SERDOV	0	0.983		

У перших двох рядках таблиці подано інформацію про об'єкт QUEUE з номером 1. Вона свідчить, що найбільша довжина черги за час моделювання становила 6 одиниць; у момент завершення моделювання черги не було; загальна кількість входжень у чергу – 169; кількість входжень з нульовим часом очікування – 45; середня довжина черги – 0,983; середній час очікування з урахуванням усіх входжень у чергу – 5,819, а без урахування „нульових” входжень – 7,931. У третьому рядку таблиці виведено вміст комірки зберігання SerDov, яка містить значення СЧА QA черги QUEUE 1, яке дорівнює 0,983, тобто ще раз (для порівняння) визначає середню довжину черги, автоматично виведену раніше в інформації про об'єкт QUEUE з номером 1.

3.8.2. Статистичні таблиці

Для отримання щільності розподілу, її інтегральних відносних частот, середнього та стандартного відхилення деяких аргументів, якими можуть бути СЧА (наприклад, часу перебування транзакта в моделі або затримки в її окремих частинах, довжин черг, вмісту БКП тощо), використовують *статистичні таблиці* TABLE і QTABLE.

Команда опису таблиці TABLE має такий формат:

Name TABLE A,B,C,D

Команда визначає аргумент, а також кількість та ширину частотних інтервалів (класів). Мітка Name визначає ім'я таблиці. Операнд A задає аргумент таблиці – елемент даних, частотний розподіл якого буде табулюватися. Операнд може бути іменем, виразом у дужках або СЧА. Операндом B задають верхню межу першого інтервалу. Операнд C задає ширину частотного інтервалу – різницю

між верхньою та нижньою межею кожного частотного класу. Перший з інтервалів має ширину від $-\infty$ до величини, зазначеної в полі В, включно, другий – це проміжок $(B, B+C]$ і так далі. Останній інтервал містить усі значення, більші від останньої межі.

Операнди В і С можуть бути числами або рядками. Операнд D задає кількість частотних інтервалів (додатне ціле число).

Для реєстрації даних транзакт повинен увійти до блока TABULATE з тим самим іменем таблиці, яке визначено у блоці TABLE. Блок записують у тому місці моделі, яке відповідає досліджуваному об'єктові.

Блок TABULATE має такий формат:

```
TABULATE A,[B]
```

Операндом А задає ім'я таблиці, в яку табулюється значення аргумента. Операнд В (ваговий коефіцієнт) визначає кількість одиниць, які мають заноситися до того частотного інтервалу, куди потрапило значення аргумента. Якщо операнд В не використовується, то за замовчуванням ваговий коефіцієнт дорівнює 1.

Операнди А і В можуть бути іменем, виразом у дужках, СЧА або СЧА*параметром. Крім того, операнд А може бути лише додатним цілим числом, а операнд В – додатним числом. Наприклад:

```
ChasRem TABLE P$Rest,8.5,1.5,10
```

```
.....
```

```
TABULATE ChasRem
```

Оператор TABLE описує таблицю з іменем ChasRem. Аргументом таблиці є СЧА P\$Rest з верхньою межею першого інтервалу 8,5, шириною 1,5 і кількістю інтервалів 10. Якщо значення аргумента потрапляє до певного частотного класу, то його частота збільшується на 1, оскільки операнд В у блоці TABULATE відсутній.

Під час входження транзакта до блока TABULATE коректуються поточні СЧА таблиці: лічильник входжень у таблицю TC, середнє значення табульованого аргумента ТВ і середнє квадратичне відхилення TD. Зібрана у таблиці статистика виводиться у стандартному звіті.

Крім таблиці TABLE, використовують Q-таблиці QTABLE, які слугують лише для отримання розподілу часу перебування транзакта в черзі. Формат опису таблиці QTABLE такий самий, як і TABLE:

```
Name QTABLE A,B,C,D
```

Відмінність полягає лише в тому, що *операнд А задає ім'я черги*. Призначення операндів В, С і D таке саме, як і в команді опису TABLE. Операнд D може бути нулем або додатним числом.

Наприклад:

```
Wtime QTABLE Dovzh,10.5,2.5,15
```

Тут Dovzh – ім'я черги до пристрою, а не СЧА, як у випадку використання TABLE. Під час проходження транзакта через блоки QUEUE і DEPART його час очікування фіксується, і до лічильника частотного інтервалу таблиці, до

якого потрапило це значення, додається 1. Одночасно в таблиці накопичується інформація для обчислення середнього значення і середнього квадратичного відхилення часу очікування.

Варто звернути увагу, що в разі використання QTABLE інформація до таблиці заноситься автоматично під час входження транзакта до блоків QUEUE і DEPART і жодних спеціальних заходів, тобто блока TABULATE, для цього не потрібно. Після завершення моделювання зібрана в таблиці інформація також виводиться у стандартному звіті GPSS.

Застосуємо оператори TABLE і QTABLE для отримання додаткової інформації про статистику очікування та перебування в системі для СМО, розглянутої в прикладі 3.6.

Приклад 3.7. *Визначення розподілу часу перебування замовлення в черзі та розподілу часу перебування в системі для двоканальної СМО з очікуванням:*

```

;   Приклад 3.7.
;   Задання БКП, опис таблиць
System      STORAGE  2
Wtime      QTABLE   1,0,3,5
Ttime      TABLE   M1,5,5,9
;   Сегмент імітування надходження та обслуговування
;   замовлень і збору статистики очікування
      GENERATE      (Uniform(3,2,10))
      QUEUE         1
      ENTER         System
      DEPART        1
      ADVANCE       (Uniform(5,2,20))
      LEAVE         System
      TABULATE      Ttime
      TERMINATE
;   Сегмент задання часу моделювання
      GENERATE 1000
      TERMINATE 1
;   START 1

```

Наведемо фрагмент стандартного звіту, який містить таблицю Wtime розподілу часу перебування в черзі й таблицю Ttime розподілу часу перебування замовлення в системі.

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
WTIME	5.819	7.706		0		
			- 0.000		45	26.63
			0.000 - 3.000		36	47.93
			3.000 - 6.000		30	65.68
			6.000 - 9.000		24	79.88
			9.000 -		34	100.00

TTIME	16.681	8.904	0		
		0.000 - 5.000		10	5.99
		5.000 - 10.000		25	20.96
		10.000 - 15.000		42	46.11
		15.000 - 20.000		49	75.45
		20.000 - 25.000		19	86.83
		25.000 - 30.000		8	91.62
		30.000 - 35.000		4	94.01
		35.000 - 40.000		3	95.81
		40.000 -		7	100.00

Оператор QTABLE визначення таблиці з іменем Wtime розбиває вісь часу на п'ять частотних інтервалів. Оператор TABLE визначає таблицю з іменем Ttime, яка містить дев'ять частотних інтервалів. Аргументом цієї таблиці є СЧА M1 – час перебування транзакта в моделі. У цій моделі значення СЧА M1 одночасно буде часом перебування замовлення в СМО, якщо занесення інформації до таблиці здійснювати перед виходом транзакта з моделі. Тому блок TABULATE, який заносить інформацію до таблиці Ttime, розташований перед блоком TERMINATE.

Перші рядки обидвох таблиць містять: ім'я таблиці, середнє значення табульованого аргумента (MEAN), його середнє квадратичне відхилення (STD.DEV.), а також значення кількості транзактів, які очікують виконання спеціальної умови, яка залежить від стану даної таблиці (RETRY).

Кожен з наступних рядків містить інформацію про один з частотних інтервалів. У стовпці RANGE розташовані нижня та верхня межі частотного класу; FREQUENCY – сумарна частота, яка формується під час потрапляння табульованого аргумента до зазначених меж; CUM.% – величина накопиченої частоти у відсотках до загальної кількості значень табульованого аргумента.

Зауважимо, що для таблиці Wtime верхня межа першого частотного інтервалу дорівнює нулю, що дає змогу окремо отримати дані для замовлень з нульовим часом очікування. Справді, кількість замовлень з нульовим часом очікування, визначена таблицею Wtime, дорівнює 45, стільки ж, що й у таблиці, яка містить інформацію про об'єкт QUEUE з номером 1 (див. приклад 3.6).

Приклад 3.8. *Визначення розподілу інтервалу надходження замовлень для потоку замовлень, який є сумою п'яти потоків з інтервалами між подіями, розподіленими за узагальненим законом Ерланга 2-го порядку.*

; Приклад 3.8.

; **Визначення арифметичних змінних, опис таблиці**

T1 FVARIABLE 0

T2 FVARIABLE 0

CHAS TABLE X\$T2,0,0.5,11

; Імітування сумарного потоку й табулювання розподілу

; часу між транзактами цього потоку

GENERATE (Exponential(1,0,5)+ Exponential(1,0,5))

```

TRANSFER ,VSI
GENERATE (Exponential(1,0,5)+ Exponential(1,0,5))
TRANSFER ,VSI
GENERATE (Exponential(1,0,5)+ Exponential(1,0,5))
TRANSFER ,VSI
GENERATE (Exponential(1,0,5)+ Exponential(1,0,5))
TRANSFER ,VSI
GENERATE (Exponential(1,0,5)+ Exponential(1,0,5))
TRANSFER ,VSI
VSI SAVEVALUE T2,(AC1-X$T1)
TABULATE CHAS
SAVEVALUE T1,AC1
TERMINATE
; Сегмент задання часу моделювання
GENERATE 1000
TERMINATE 1
START 1

```

Наведемо фрагмент стандартного звіту, який містить таблицю CHAS розподілу часу між замовленнями сумарного потоку.

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
CHAS	2.032	1.782		0		
	0.000 -	0.500			101	20.61
	0.500 -	1.000			78	36.53
	1.000 -	1.500			62	49.18
	1.500 -	2.000			51	59.59
	2.000 -	2.500			37	67.14
	2.500 -	3.000			49	77.14
	3.000 -	3.500			22	81.63
	3.500 -	4.000			20	85.71
	4.000 -	4.500			14	88.57
	4.500 -				56	100.00

Оператор TABLE визначає таблицю з іменем CHAS, яка містить 11 частотних інтервалів. Аргументом цієї таблиці є СЧА X\$T2, обчислений як різниця між значенням абсолютного модельного часу AC1 у момент потрапляння чергового транзакта до блока SAVEVALUE з міткою VSI і значенням арифметичної змінної T1, у якій записане відповідне значення AC1 для попереднього транзакта. Сумування потоків відбувається шляхом скерування всіх транзактів до блока SAVEVALUE з міткою VSI блоком TRANSFER, який працює в режимі безумовного переходу.

Інтенсивність сумарного потоку, як відомо, дорівнює сумі інтенсивностей потоків-доданків. Тому в цьому випадку інтенсивність сумарного потоку дорівнює $5 \cdot 0,1 = 0,5$, а отже, середнє значення інтервалу часу між замовленнями сумарного потоку має наближатися до двох одиниць модельного часу. Це

узгоджується з результатами моделювання, оскільки перший рядок таблиці CHAS містить середнє значення СЧА X_{ST2} , яке дорівнює 2,032. Відповідне значення середнього квадратичного відхилення табульованого аргумента становить 1,782.

3.9. Переведення багатоканального пристрою до недоступного стану та відновлення доступності

Недоступність БКП моделюють за допомогою блока SUNAVAIL (S означає БКП, UNAVAIL – недоступний). Формат блока такий:

SUNAVAIL A

Операнд A – ім'я або номер БКП; може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

SUNAVAIL Syst

Коли транзакт потрапляє до цього блока, то БКП Syst стає недоступним. Якщо в час переведення до недоступного стану в БКП перебували транзакти, тобто поточний вміст БКП не дорівнював нулю, то обслуговування цих транзактів продовжується, поки поточний вміст БКП не буде дорівнювати нулю. Транзакти, які намагаються зайняти БКП у час перебування його в недоступному стані, не заходять до блока ENTER і заносяться до списку затримки БКП.

Перебування в недоступному стані триває до того часу, поки транзакт не увійде до блока SAVAIL. Формат блока:

SAVAIL A

Операнд A – ім'я або номер БКП; може бути таким, як і операнда A блока SUNAVAIL.

Якщо в момент відновлення доступності БКП у його списку затримки були транзакти, їм надається можливість зайняти БКП відповідно до дисципліни „first-fit-with-skip” (перший придатний з перервою). Транзакти, яким буде відмовлено займання БКП, залишаються в списку затримки.

Приклад 3.9. Застосування блоків SUNAVAIL і SAVAIL для імітування недоступності БКП.

```

; Приклад 3.9.
; Задання БКП
System STORAGE 4
; Сегмент імітування надходження та обслуговування
; замовлень
GENERATE ,,4
ENTER System
ADVANCE 15
LEAVE System
TERMINATE
; Сегмент імітування недоступності
GENERATE 10,,1
SUNAVAIL System
ADVANCE 7
SAVAIL System
TERMINATE 1

```

3.9. Переведення багатоканального пристрою до недоступного стану та відновлення доступності

START 1

Блок GENERATE першого сегмента генерує в момент $t=0$ чотири транзакти, які займають БКП System ємністю 4, визначеною командою STORAGE.

У час $t=10$ блок GENERATE другого сегмента генерує транзакт, який, заходячи до блока SUNAVAIL, переводить БКП System у недоступний стан.

Наведемо результати моделювання.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	17.000	10	0	1

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
1	GENERATE	4		0	0			
2	ENTER	4		0	0			
3	ADVANCE	4		0	0			
4	LEAVE	4		0	0			
5	TERMINATE	4		0	0			
6	GENERATE	1		0	0			
7	SUNAVAIL	1		0	0			
8	ADVANCE	1		0	0			
9	SAVAIL	1		0	0			
10	TERMINATE	1		0	0			

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SYSTEM	4	4	0	4	4	1	3.529	0.882

БКП System у недоступному стані перебував 7 одиниць модельного часу. У момент $t=17$ блоком SAVAIL була відновлена його доступність. Час обслуговування транзактів дорівнює 15 одиницям. Результати моделювання показують, що й у період недоступності обслуговування чотирьох транзактів, які перебували в БКП, тривало ще 5 одиниць модельного часу й завершилось у момент $t=15$. Всі чотири обслужені транзакти увійшли до блока TERMINATE першого сегмента.

Потрібно мати на увазі, що недоступність лише забороняє надходження транзактів до БКП. Обслуговування ж транзактів, які ще не завершили обслуговування в момент відновлення доступності, продовжується й після відновлення доступності. Отже, здійснити імітування виходу БКП з ладу, що призвело б до втрати всіх транзактів, які перебували на обслуговуванні, за допомогою блоків SUNAVAIL і SAVAIL неможливо.

3.10. Моделювання виходу з ладу багатоканального пристрою

Імітацію виходу з ладу БКП, коли всі транзакти, що перебували в БКП на обслуговуванні, втрачаються, можна здійснити з використанням блоків DISPLACE, SUNAVAIL і SAVAIL.

Приклад 3.10. Багатоканальна СМО з відмовами без черги. СМО має обмежену надійність. У момент виходу СМО з ладу замовлення, що перебувають на обслуговуванні, втрачаються, і нові замовлення на обслуговування не приймаються.

```

; Приклад 3.10.
; Задання БКП і булевих змінних
Kan EQU 7
Zaym EQU 2
Syst STORAGE 7
Per1 BVARIABLE SV$Syst'AND'((7-S$Syst)>=Zaym)
Per2 BVARIABLE SE$Syst
; Сегмент імітування надходження й обслуговування замовлень
GENERATE 4,,7
Mit12 TEST E BV$Per1,1,Mit1
SAVEVALUE KPovt,(INT(Kan/Zaym))
ENTER Syst,Zaym
; Облік номерів транзактів, які зайняли БКП
ASSIGN KPovt,X$KPovt
Mit5 TEST E X*KPovt,0,Mit4
SAVEVALUE P$KPovt,XN1
TRANSFER ,Mit6
Mit4 LOOP KPovt,Mit5
;
Mit6 ADVANCE 10
LEAVE Syst,Zaym
; Облік номерів транзактів, які звільнили БКП
ASSIGN KPovt,X$KPovt
Mit8 TEST E X*KPovt,XN1,Mit7
SAVEVALUE P$KPovt,0
TRANSFER ,Mit9
Mit7 LOOP KPovt,Mit8
;
Mit9 TERMINATE 1 ; Обслужені замовлення
Mit1 TERMINATE 1 ; Втрачені замовлення
; Сегмент імітування неполадок
GENERATE ,,1
Mit2 ADVANCE 15 ; Інтервал відмов
SUNAVAIL Syst ; Перехід у недоступність

```

```

; Вилучення транзактів з БКП
      ASSIGN      KPovt,X$KPovt
Mit10 TEST NE     X*KPovt,0,Mit11
      DISPLACE   X*KPovt,Mit3
      SAVEVALUE  P$KPovt,0
Mit11 LOOP      KPovt,Mit10
;
      TEST E     BV$Per2,1 ; БКП порожній?
      ADVANCE   2          ; Час ремонту
      SAVAIL    Syst ; Перехід у стан доступності
      TRANSFER  ,Mit2
Mit3  LEAVE     Syst,Zaym
      TERMINATE 1          ; Вилучені з БКП замовлення
      START    7

```

Блок GENERATE з інтервалами 4 одиниці модельного часу генерує сім транзактів. Наступний блок TEST перевіряє БКП Syst на доступність і наявність у ньому вільних каналів, достатніх для задоволення запиту. Якщо булева змінна Per1 дорівнює 1, то транзактові дозволяють входження до БКП Syst. Проте перед тим обчислюється та заноситься ціле число $\text{INT}(\text{Kan}/\text{Zaym}) = \text{INT}(7/2) = 3$ у комірку зберігання з іменем KPovt, яке визначає, скільки транзактів може одночасно перебувати в БКП.

Після виходу транзакта з блока ENTER починає працювати сегмент обліку номерів транзактів, які зайняли БКП. Блоком ASSIGN до параметра з іменем Kpovt (параметр циклу) заноситься число, яке записане в комірці зберігання з іменем Kpovt. Далі в циклі, тіло якого починається з блока TEST з міткою Mit5 і закінчується блоком LOOP з міткою Mit4, відшукується вільне місце у списку для запису номера транзакта. Вільне місце визначається блоком TEST з умови, що дорівнює нулеві значення якоїсь одної з трьох комірок зберігання X1, X2 або X3 (за кількістю транзактів, які одночасно перебувають в БКП). Така комірка завжди є і блоком SAVEVALUE до неї записується номер транзакта, який зайняв БКП. Оскільки комірок зберігання, значення яких дорівнюють нулеві, може бути кілька (особливо на початковій стадії роботи моделі), то після запису номера транзакта здійснюється вихід з циклу.

Транзакт скеровується до блока ADVANCE з міткою Mit6, заходить до нього, імітуючи обслуговування замовлення. Після обслуговування транзакт звільняє БКП Syst, пройшовши блок LEAVE.

Починає працювати сегмент обліку номерів транзактів, які звільнили БКП. Його робота аналогічна розглянутій раніше роботі сегмента обліку номерів транзактів, які зайняли БКП. Відмінність полягає в тому, що відшукується комірка зберігання X1, X2 або X3, яка містить номер обслуженого транзакта. У знайдену комірку записується число нуль, тобто обслужений транзакт знімається з обліку. Оскільки номер записується лише в одну комірку, то після запису до неї нуля здійснюється вихід з циклу, який

починається з блока TEST з міткою Mit8 і закінчується блоком LOOP з міткою Mit7.

Тепер розглянемо роботу моделі у випадку виникнення неполадок у БКП. Транзакт, який імітує неполадки, після затримки блоком ADVANCE сегмента імітування неполадок, заходить до блока SUNAVAIL, і цей блок переводить БКП Syst у недоступний стан.

Далі починає працювати сегмент вилучення транзактів з БКП. У циклі, також організованому за допомогою блока LOOP, переглядається список номерів транзактів, які перебувають в БКП. Якщо значення якоїсь з комірок зберігання X1, X2 або X3 не дорівнює нулеві, то це означає, що в ній записано номер транзакта, який перебуває в даний момент у БКП. Блоком DISPLACE цей транзакт передається до блока LEAVE з міткою Mit3, залишає БКП і виводиться з моделі.

Після вилучення з БКП всіх транзактів, транзакт, який є ініціатором неполадок, заходить до блока TEST, розташованого після блока LOOP з міткою Mit11. Оскільки БКП Syst порожній (булева змінна Per2=0), то блок TEST не затримує його і починається імітування відновлення працездатності БКП.

Нижче наведено фрагмент стандартного звіту GPSS з результатами моделювання, а також графік залежності кількості зайнятих каналів від часу, отриманий інструментальними засобами GPSS World з використанням СЧА S\$\$Syst.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	32.000	32	0	1

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES AVL.	AVE.C.	UTIL.
SYST	7	7	0	6	12	0	2.625 0.375

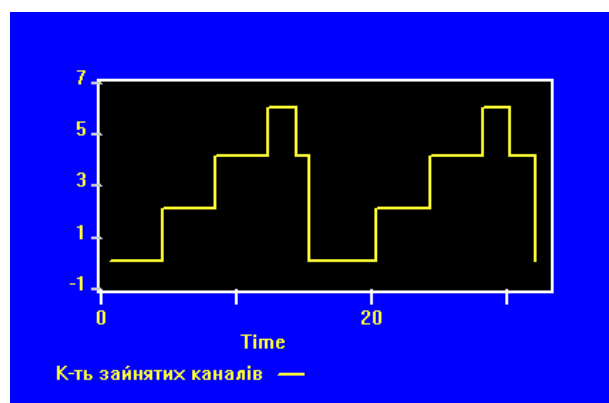


Рис. 3.1. Графік залежності кількості зайнятих каналів від часу для моделі прикладу 3.10

З графіка та табл. 3.2 випливає, що обслужено лише два замовлення

(перший і п'ятий транзакти, створені першим блоком GENERATE). П'ять замовлень не обслужені: четвертий транзакт надійшов, коли в БКП виникли неполадки, а другий і третій, шостий і сьомий транзакти втрачені: в моменти виникнення неполадок ($t=15$ і $t=32$ відповідно) вони перебували в БКП.

Таблиця 3.2

Результати моделювання прикладу 3.10

Події	Номери транзактів/ Модельний час						
	1	2	3	4	5	6	7
Вхід до моделі	4	8	12	16	20	24	28
Займання БКП	4	8	12		20	24	28
В БКП транзакти	1	1, 2	1,2,3		5	5, 6	5,6,7
Вилучення		15	15			32	32
Звільнення БКП	14	15	15		30	32	32
Вихід з моделі	14	15	15	16	30	32	32

Коефіцієнт використання (UTIL.) БКП Syst згідно з результатами стандартного звіту дорівнює 0,375. Його обчислено як відношення площі фігури, обмеженої графіком (рис. 3.1) і осями координат, до площі прямокутника зі сторонами 32 (час моделювання) і 7 (ємність БКП).

3.11. Логічні ключі

До апаратних об'єктів належать також *логічні перемикачі* (або ключі), які можуть перебувати у двох станах:

- увімкнено (ON або 1);
- вимкнено (OFF або 0).

Залежно від стану логічного ключа змінюється напрям руху транзактів.

Логічний ключ моделюють за допомогою блока LOGIC, який має такий формат:

LOGIC X A

Операнд A – ім'я або номер ключа. Може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Стан логічного ключа залежить від таких значень логічного оператора X:

- S – логічний ключ, заданий операндом A, вмикається;
- R – логічний ключ, заданий операндом A, вимикається;
- I – логічний ключ інвертується, тобто його стан змінюється на протилежний, наприклад, якщо був увімкнений, то стане вимкненим.

Перевірку стану логічного ключа здійснюють за допомогою блоків GATE або TEST, які мають такий самий формат, як і під час перевірки стану ОКП або БКП.

Блок GATE має формат:

GATE X A,[B]

Операнд A – ім'я або номер ключа, який перевіряється. Може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Операнд B – мітка блоку, до якого буде скеровано транзакт у випадку невиконання умови, заданої умовним оператором X.

Умовний оператор може набувати значень:

- LS – дорівнює 1, якщо логічний ключ, заданий операндом A, увімкнений; 0 – якщо вимкнений;
- LR – дорівнює 1, якщо логічний ключ, заданий операндом A, увімкнений; 0 – якщо вимкнений.

Формат та умовні оператори блока TEST такі самі, як і для ОКП або БКП. У булевій змінній використовують лише СЧА LS. Якщо в булевій змінній записати LR, то виникне помилка “Припускається фактор виразу”.

3.12. Блоки для роботи зі списками користувача

Під час руху в моделі транзакти можуть бути заблоковані, наприклад, блоками GATE або TEST. Якщо заблоковані транзакти перебувають у списку поточних подій, то у випадку, коли їх багато, планувальник витрачає велику кількість часу на перегляд списку з метою вибору чергового транзакта для просування. Для економії машинного часу заблоковані транзакти доцільно заносити до *списків користувача* й залишати їх там до того часу, поки не будуть виконані умови, які дозволять подальший рух цих транзактів. Крім того, розташування транзактів у списках користувача дає змогу організувати різні дисципліни черг, які відрізняються від дисципліни „першим прийшов – першим обслуговується” (FIFO), реалізованої у списку поточних подій.

Список користувача є деяким буфером, до якого можуть тимчасово заноситися транзакти, виведені зі списку поточних подій. *На відміну від списків поточних і майбутніх подій транзакти вводяться до списку користувача й виводяться з нього не автоматично, а згідно з рішенням користувача відповідно до логіки моделі за допомогою спеціальних блоків.*

Для введення транзактів до списку користувача слугує блок LINK (ввести до списку), який може бути використаний у двох режимах: умовному й безумовному. Обмежимося розглядом лише безумовного режиму, в якому блок LINK має такий формат запису:

[ім'я] LINK A,B

Операнд A задає ім'я або номер списку користувача, до якого автоматично заноситься транзакт після входження до блока LINK.

Операнд B визначає, в яке місце списку користувача треба внести активний транзакт. Допустимі значення:

- FIFO – транзакт передається в кінець списку;
- LIFO – транзакт передається на початок списку;
- PR – транзакти впорядковуються за спаданням пріоритету;
- P – транзакти розташовуються позаду тих транзактів, значення відповідного параметра яких менше (у порядку зростання значення параметра);
- M1 – транзакти розташовуються в порядку зростання відносного часу перебування в моделі.

У полі операнда B можна використовувати й інші СЧА, крім зазначених вище СЧА транзактів: арифметичну змінну, функцію, а також вираз у дужках. У цьому випадку здійснюється обчислення вказаного СЧА для активного транзакта й для решти транзактів, які вже перебувають у списку користувача, починаючи з початку черги. Після цього відбувається впорядкування транзактів у списку користувача за спаданням обчисленого значення.

Наприклад, блок

LINK 2,FIFO

заносить транзакти в кінець списку користувача номер 2 у порядку їхнього

надходження до блока. Блок

LINK Buf,P\$Kor

заносить транзакти в список користувача з іменем Buf, впорядковуючи їх за зростанням значення параметра з іменем Kor.

Умови, унаслідок виконання яких транзакт заноситься до списку користувача, в безумовному режимі перевіряються засобами, передбаченими розробником моделі. Наприклад, скерувати транзакт до списку користувача за умови зайнятості ОКП можна так, як показано нижче.

```

.....
GATE NU   Syst,Wait
SEIZE     Syst
.....
Wait      LINK       Buf,FIFO
.....

```

Якщо ОКП Syst зайнятий, то блок GATE не допускає транзакта до блока SEIZE, а скерує його до блока LINK з іменем Wait, і транзакт заноситься у кінець списку користувача з іменем Buf.

У попередньому прикладі припускалося, що список користувача необмежений, тобто до нього можна занести будь-яку кількість транзактів. У реальних системах список користувача можна використовувати для імітування, наприклад вхідного накопичувача, ємність якого, зазвичай, обмежена. Це обмеження можна реалізувати так.

```

Dov      EQU       5
.....
GATE NU   Syst,Wait
SEIZE     Syst
.....
Wait      TEST L   CH$Buf, Dov,Term
          LINK     Buf,FIFO
.....
Term      TERMINATE
.....

```

Якщо ОКП Syst зайнятий, то блок GATE не допускає транзакта до блока SEIZE, а скерує його до блока TEST з іменем Wait, який розташований перед блоком LINK. Якщо поточна довжина списку користувача з іменем Buf менша від заданої (Dov), то транзакт проходить до списку користувача; в іншому випадку він скерується до блока TERMINATE з міткою Term.

Кожен список користувача має такі СЧА: СН – поточна довжина списку; СА – середня довжина списку; СМ – максимальна довжина списку; СС – загальна кількість транзактів, які потрапили до списку; СТ – середній час перебування транзакта в списку.

Для виведення одного або кількох транзактів зі списку користувача й занесення їх назад до списку поточних подій слугує блок UNLINK (вивести зі

списку), який має такий формат:

[ім'я] UNLINK X A,B,C,[D],[E],[F]

Операнд А задає ім'я або номер списку користувача.

Операнд В – мітка блока, до якого переходять виведені зі списку транзакти.

Операнд С вказує кількість транзактів, які виводяться, або ключове слово ALL для виведення всіх транзактів, які є в списку. За замовчуванням, тобто коли не використовується операнд С, береться ALL.

Операнди D і E разом з умовним оператором X визначають спосіб і умови виведення транзактів зі списку користувача. Значення оператора X такі самі, як і у блоці TEST. У випадку, коли умовний оператор X потрібно використовувати, але він не вказаний, то за замовчуванням він набуває значення E (рівність). Якщо операндів D та E немає, то не використовують також умовний оператор X. У цьому випадку транзакти виводяться з початку списку, а кількість таких транзактів визначається обов'язковим операндом С.

Операнд D може бути:

- булевою змінною;
- номером параметра транзакта;
- ключовим словом VACK.

Якщо операнд D є булевою змінною, то операнд E і оператор X не використовуються. Булева змінна підраховується відносно транзакта зі списку користувача. Якщо результат не нуль, тобто умова виведення виконується, то транзакт виводиться. Кількість транзактів, які виводяться, визначає операнд С. Однак виведено може бути і менше, ніж вказано операндом С: за кількістю результатів обчислення булевої змінної, які не дорівнюють нулю. Крім того, і транзактів у списку користувача може бути менше, ніж зазначено операндом С.

Якщо операнд D вказує ключове слово VACK, то операнд E й умовний оператор X також не використовуються, а транзакти виводяться з кінця списку в кількості, яка визначається обов'язковим операндом С.

Якщо операнд D не булева змінна і не ключове слово VACK, то мають бути вказані операнд E та умовний оператор X. Операнд D підраховується відносно транзакта зі списку користувача й використовується як номер параметра, значення якого порівнюється з результатом обчислення операнда E.

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. У цьому випадку кількість транзактів, які виводяться, також визначає операнд С.

Операндом F вказується ім'я блока, куди переходить транзакт, який виходить з блока UNLINK, якщо зі списку користувача не виведено жодного транзакта. Якщо операнд С не використовується, то транзакт, який виходить, переходить до наступного блока незалежно від кількості виведених транзактів.

Наприклад, блок

UNLINK 2, Ter, 1

виводить зі списку користувача з номером 2 один транзакт з початку списку і скеровує його до блока з міткою Ter. Блок

UNLINK Buf, Sat, 1, BACK

виводить зі списку користувача з іменем Buf один транзакт з кінця списку і скеровує його до блока з міткою Sat. Блок

UNLINK E P\$Obl, Mit2, ALL, Kond, P\$Kond, Mit3

виводить зі списку користувача, номер якого записаний у параметрі Obl транзакта, який ініціює виведення, і скеровує до блока з іменем Mit2 всі транзакти, значення параметра Kond яких дорівнює значенню однойменного параметра транзакта, який ініціює виведення. Якщо таких параметрів у списку немає, то транзакт, який ініціює виведення, буде скерований до блока з іменем Mit3, а в іншому випадку – до наступного блока.

Зазначимо такі особливості виконання блока UNLINK. По-перше, якщо операнди D і E містять посилання на СЧА транзактів, то операнд C обчислюється відносно транзактів у списку користувача, а операнд E – відносно активного транзакта. По-друге, після виведення транзактів зі списку планувальник продовжує або починає просування транзакта з найвищим пріоритетом, а якщо вони рівні, то віддає перевагу транзактові-ініціаторові виведення.

Розглянемо детальніше на інших прикладах механізм роботи блоків LINK та UNLINK.

Приклад 3.11. Багатоканальна СМО з очікуванням транзактів у списку користувача.

```

;   Приклад 3.11.
Syst STORAGE      2
;
      GENERATE      (Uniform(3,2,10))
      GATE SNF      Syst,Wait
Vhid  ENTER        Syst
      ADVANCE      (Uniform(5,2,20))
      LEAVE        Syst
      UNLINK       Buf,Vhid,1
      TERMINATE
Wait  LINK         Buf,FIFO
;
      GENERATE      1000
      TERMINATE    1
      START        1

```

Якщо БКП Syst не заповнений, то блок GATE пропускає активний транзакт до блока ENTER, і в БКП займається один канал. Якщо ж БКП заповнений, то блок GATE скеровує транзакт до блока LINK з іменем Wait, а той заносить транзакт на кінець списку користувача з іменем Buf, який

моделює чергу до БКП. Кожен транзакт, залишаючи БКП після завершення обслуговування, звільнює один канал, проходить блок UNLINK і виводить один транзакт з початку списку (якщо список не порожній), скеровуючи його до блока з іменем Vhid для займання каналу в БКП.

Варто зазначити, що блоки QUEUE і DEPART для реєстрації статистики про чергу не використовуються, оскільки майже всі такі самі дані можна отримати зі статистики стандартного звіту про список користувача. Нижче наведено порівняння цих даних для моделі прикладу 3.11 і для випадку використання блоків QUEUE і DEPART.

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
BUF	0	0	0.983	124	6	7.931

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME
1	6	0	169	45	0.983	5.819

AVE.(-0)	RETRY
7.931	0

Для того щоб змінити дисципліну обслуговування на „пізніше прибув – раніше обслужений” досить або замінити FIFO у полі операнда В блока LINK на LIFO, або записати ключове слово BACK у полі D оператора UNLINK.

Розглянемо ще один приклад, який показує використання списків користувача для організації нестандартних дисциплін обслуговування.

Приклад 3.12. В одноканальній СМО з очікуванням потрібно організувати таку дисципліну обслуговування, коли пріоритет віддається замовленням з найменшим часом обслуговування.

```

;   Приклад 3.12.
      GENERATE   (Uniform(3,2,10))
      ASSIGN     Tmin,(Uniform(4,1,15))
      GATE NU    Syst,Wait
Vhid SEIZE      Syst
      ADVANCE    P$Tmin
      RELEASE    Syst
      UNLINK     Buf,Vhid,1
      TERMINATE
Wait LINK       Buf,P$Tmin
;
      GENERATE   1000
      TERMINATE  1
      START      1

```

До параметра Tmin активних транзактів у блоці ASSIGN записується випадковий час обслуговування, обчислений за допомогою вмонтованого генератора рівномірного розподілу. Якщо ОКП Syst вільний, то блок GATE пропускає транзакт до блока SEIZE, і ОКП займається на час P\$Tmin. Якщо ж у момент надходження транзакта ОКП зайнятий, то блок GATE скеровує

транзакт до блока LINK з міткою Wait, де транзакт вводиться до списку Buf із впорядкуванням за зростанням часу обслуговування, записаного в параметрі P\$Tmin. Блок UNLINK у момент звільнення ОКП виводить з початку списку транзакт з найменшим часом обслуговування, забезпечуючи тим самим задану дисципліну.

3.13. Блок PRIORITY

Блок PRIORITY присвоює транзактові 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 А 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 В визначає режим BUFFER, у ньому вказується значення "BU". З блоком PRIORITY пов'язаний СЧА PR – значення пріоритету активного транзакта.

Загалом процедура пререгляду списку поточних подій намагається просунути активний транзакт до наступного блока. Однак, якщо в полі В блока PRIORITY записано слово BU, то після присвоєння транзактові 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 транзактів, які увійшли до цього блока, то транзакт буде опрацьований процедурою перегляду знову без зміни значення модельного часу.

Наприклад: **PRIORITY 10**

Тут активному транзактові присвоюється пріоритет, який дорівнює 10.

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. Наприклад, це може бути зроблено перед входженням транзакта до блока SPLIT. Оскільки транзакт опиниться останнім у своєму класі пріоритетів, то транзакти-копії, створені блоком SPLIT, будуть розташовані у списку поточних подій безпосередньо за транзактом-оригіналом, оскільки кожна копія вводиться до списку поточних подій у кінець відповідного класу.

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.

4. СТВОРЕННЯ ТА ЕКСПЛУАТАЦІЯ МОДЕЛЕЙ В GPSS WORLD

4.1. Створення об'єкта „Модель”

4.1.1. Запуск GPSS World

Запуск GPSS World здійснюють подвійним клацанням миші на його значку, який після встановлення програми розташовується на робочому столі. Спочатку з'являється вікно Notices (Нотатки), яке містить поточну інформацію про оновлення та інші корисні відомості.

Якщо зачинити вікно Notices, то з'являється головне вікно GPSS World, яке складається з кількох компонентів. У першій стрічці (рядку заголовка) головного вікна зазначено назву вікна – GPSS World. Нижче розміщене основне меню, а ще нижче – стандартна панель інструментів.

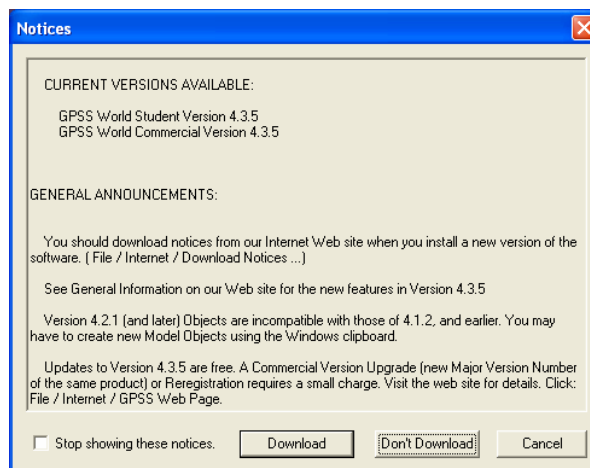


Рис. 4.1. Вікно Notices (Нотатки)

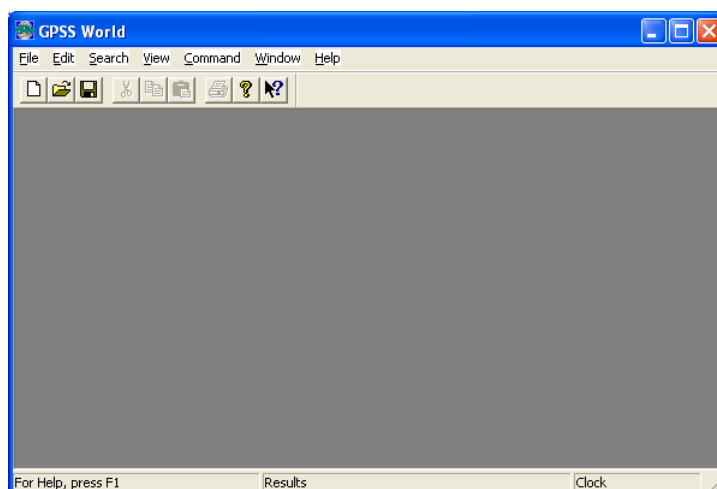


Рис. 4.2. Головне вікно GPSS World

Нижній рядок головного вікна – рядок стану системи, розділений на три частини. Ліва частина рядка стану показує підказки до пунктів меню, середня – повідомлення про помилки, а права відображає модельний час у процесі виконання моделі. Вмикання або вимикання показу модельного часу (таймера) для кожного об'єкта “Процес моделювання” здійснюється встановленням або зніманням „галочки” за адресою основного меню: View ► Simulation Clock (Вигляд ► Модельний час).

Усі об'єкти GPSS World мають меню у своїх власних вікнах. Проте значна частина дій запускається з головного вікна за допомогою основного меню (File, Edit, Search, View, Windows, Help (Файл, Редагування, Пошук, Вигляд, Вікна, Довідка)). Кожен пункт основного меню має команди підменю, більшість з яких доступні для використання лише в певних випадках. Так зроблено з метою надання допомоги у виборі правильних дій. У недоступному (вимкненому) стані команди меню відображаються сірим кольором і не можуть бути вибрані.

Є три варіанти роботи з моделлю:

- вибір за допомогою миші послідовності пунктів меню;
- натискання клавіші [Alt] для активізації меню з наступним натисканням клавіші з підкресленим символом у назві кожного пункту меню;
- використання клавіш швидкого доступу, вказаних справа від пунктів меню.

Для створення нового об'єкта “Модель” потрібно ввести оператори моделі.

4.1.2. Введення операторів моделі

Для роботи з текстом GPSS World має текстовий редактор. Виберемо File ► New (Файл ► Новий). З'явиться меню:



Рис. 4.3. Меню вибору об'єкта типу “Модель” або текстового об'єкта

Оскільки треба створити об'єкт “Модель”, то виберемо Model. Відчиниться вікно текстового редактора (рис 4.4).

У текстовому редакторі GPSS World за замовчуванням використовується шрифт Courier New. Цей шрифт полегшує вирівнювання колонок тексту моделі й особливо стандартного звіту, оскільки всі символи цього шрифту мають однакову ширину. Змінити шрифт можна, вибравши Edit ► Font (Редаг. ► Шрифт).

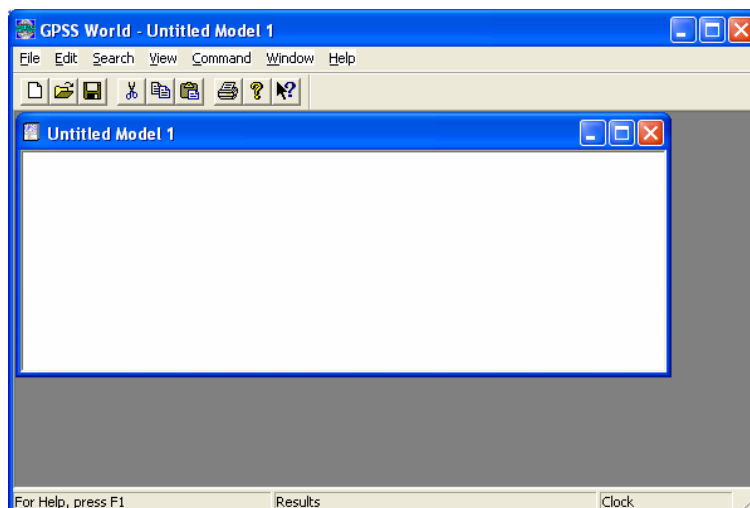


Рис. 4.4. Вікно текстового редактора

Введення всього тексту моделі у вікні текстового редактора можна здійснювати вручну. Для автоматичного введення операторів слугує вікно Edit ► Insert GPSS Blocks... (Редаг. ► Вставити блоки GPSS...). Наприклад, для введення блока GENERATE у вікні (рис. 4.5) виберемо GENERATE і клацнемо лівою клавішею миші. З'явиться вікно створення блока GENERATE (рис. 4.6) з курсором у віконечку операнда A. Ввівши всі необхідні значення операндів і, якщо потрібно, коментар у віконечку Comment, після натискання ОК, у вікні текстового редактора отримаємо рядок програми, який відповідає блоку GENERATE. Аналогічно вводяться й інші оператори, які розташовані у меню блоків GPSS.

Для внесення змін до тексту використовуються й інші опції меню Edit. Зокрема, наприклад, для відміни неправильно зробленої зміни, тобто повернення файлу до попереднього стану, використовують пункт меню Undo (Повернення) ([Ctrl]+[Z]). Команди меню Search, які також можна використовувати для редагування тексту, схожі до аналогічних в інших

ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рис. 4.5. Меню блоків GPSS

4.1. Створення об'єкта „Модель”

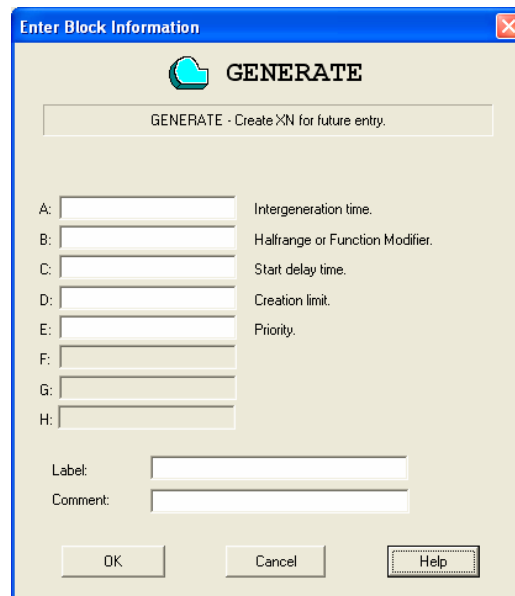


Рис. 4.6. Вікно створення блока GENERATE

текстових редакторах.

Для збереження об'єкта “Модель” треба вибрати File ► Save або Save As... (Файл ► Зберегти або Зберегти як...) ([Ctrl]+[S]). Надалі для відкриття цієї моделі або будь-якої іншої необхідно використовувати команду File►Open (Файл► Відкрити) ([Ctrl]+[O]).

Після цього можна розпочинати створення об'єкта „Процес моделювання”.

4.2. Створення об'єкта „Процес моделювання”

Об'єкт „Процес моделювання” – це відтрансльований об'єкт “Модель”. Для трансляції об'єкта “Модель” необхідно вибрати **Command►Create Simulation** (Команда ►Створити процес моделювання) ([Ctrl]+[Alt]+[S]). За цією командою транслятор GPSS перевіряє модель на наявність синтаксичних помилок. У випадку відсутності синтаксичних помилок транслятор активізує всі інтерактивні команди та вікна для того, щоб можна було стежити за станом процесу моделювання і керувати ним.

Якщо ж синтаксичні помилки в об'єкті “Модель” є, то об'єкт “Процес моделювання” не може бути створений. У цьому випадку у вікні JOURNAL (Журнал) транслятор видає список повідомлень про помилки трансляції (рис. 4.7).

Для пошуку та виправлення помилок використовують команду **Search ►Next Error** (Пошук ► Наступна помилка) ([Ctrl]+ [Alt]+[N]). Унаслідок виконання цієї команди курсор миші розташовується в рядку тексту моделі з помилкою (рис. 4.8). Одночасно помилка також описується в рядку стану головного вікна. Якщо помилок кілька, то команду **Search►Next Error** потрібно виконувати доти, поки не будуть виявлені всі помилки. Команда **Search►Previous Error** (Пошук►Попередня помилка) ([Ctrl]+[Alt]+[P]) дає змогу переглянути помилки трансляції у тексті моделі у зворотному порядку.

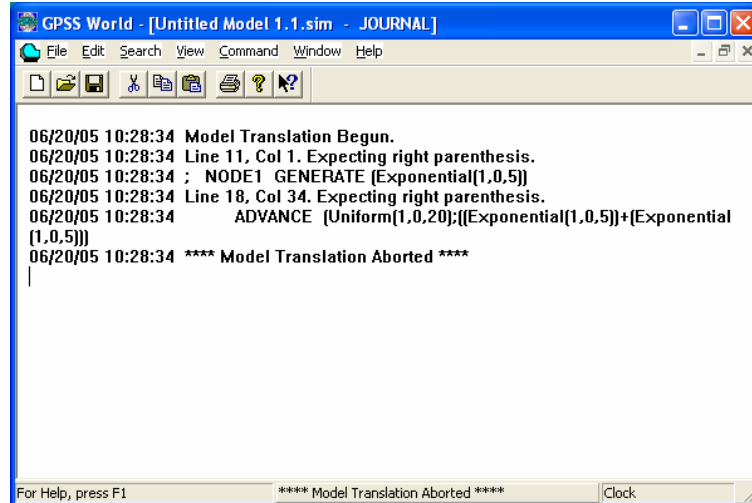


Рис. 4.7. Вікно JOURNAL зі списком повідомлень про помилки трансляції

4.2. Створення об'єкта „Процес моделювання”

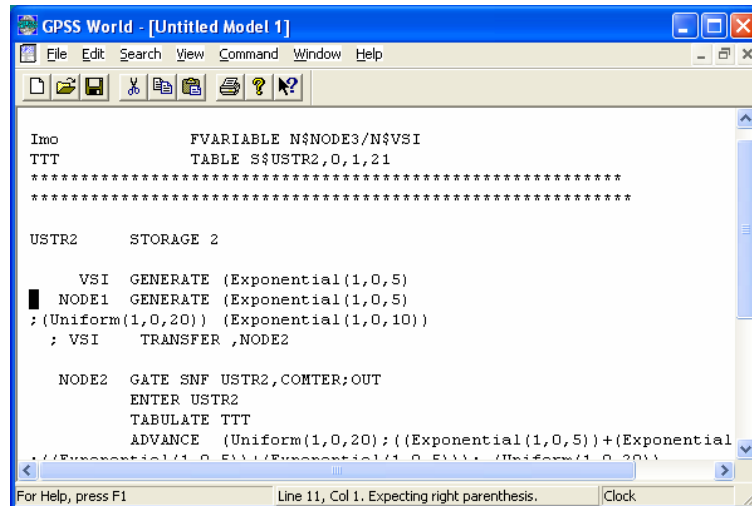


Рис. 4.8. Курсор миші в рядку тексту моделі з помилкою

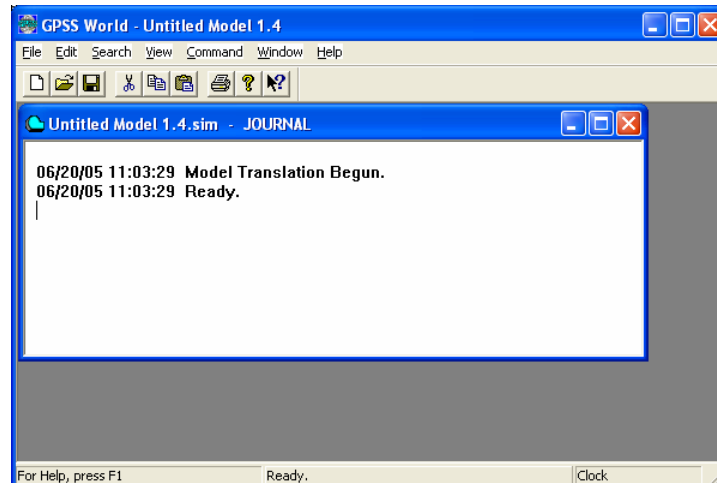


Рис. 4.9. Вікно JOURNAL зі списком повідомлень про відсутність помилок трансляції

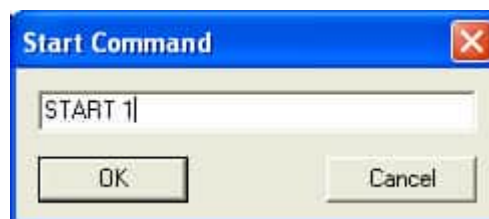


Рис. 4.10. Діалогове вікно команди Start

Після виправлення помилок треба повторно здійснити трансляцію. Її можна виконати також командою Command ► Retranslate) ([Ctrl]+[Alt]+[R]). У випадку відсутності помилок трансляції у вікні JOURNAL з'являється повідомлення про це (рис. 4.9).

Тепер можна запустити процес моделювання. Для цього потрібно вибрати Command ► Start (Команда ► Старт). З'явиться діалогове вікно (рис. 4.10), в якому є можливість замінити 1 на потрібне число, яке визначає значення лічильника завершень, наприклад на 50 або 50,NP (NP – стандартний звіт не виводиться). Після натискання ОК почнеться виконання моделі.

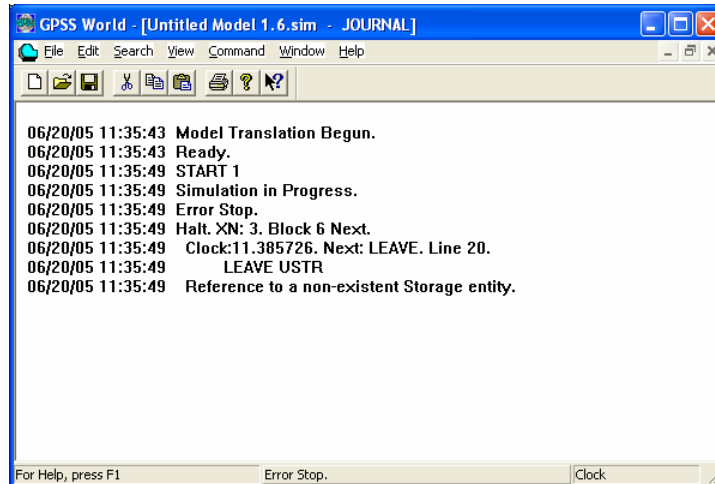


Рис. 4.11. Вікно JOURNAL з повідомленням про помилку виконання

У процесі виконання моделі можлива зупинка внаслідок помилки. У вікні JOURNAL про це також видається повідомлення (рис. 4.11). Оскільки в цьому випадку транслятор не створює списку помилок виконання, то для пошуку помилки потрібно використати команду Search ► Go To Line... (Пошук ► Перейти до рядка...) ([Ctrl]+[Alt]+[G]). Номер стрічки, яку потрібно знайти, вказаний у повідомленні про помилку виконання (див. рис. 4.11, Line 20).

4.3. Команди GPSS World

4.3.1. Інтерактивні оператори

Після успішного завершення початкової трансляції моделі створюються:

- послідовність блоків;
- реєстраційний список процедур;
- список команд.

Все це передається об'єктові “Процес моделювання”. Послідовність блоків і список команд зберігаються окремо.

Після початкової трансляції моделі створеному об'єктові “Процес моделювання” можна передавати будь-які оператори моделі. Такі оператори називають *інтерактивними*. Це можуть бути:

- команди;
- визначення Plus-процедур;
- блоки.

Послідовність блоків, створена під час початкової трансляції, залишається незмінною до наступної повторної трансляції. Інтерактивні блоки не вводяться у цю послідовність. Вони транслюються незалежно і передаються для виконання об'єктові “Процес моделювання”. Для цього після їхнього надходження створюється тимчасовий блок, який використовують у режимі ручного моделювання, а потім знищують.

Інтерактивні блоки називаються *операторами ручного режиму моделювання*. Лише блоки GENERATE не можуть використовуватися у цьому режимі. Замість них для створення транзактів можуть бути використані блоки SPLIT.

Під час початкової трансляції моделі відтрансльовані процедури заносяться до реєстраційного списку процедур. Коли об'єкт “Процес моделювання” отримує інтерактивний оператор процедури, він приєднує відтрансльовану процедуру до бібліотеки процедур процесу моделювання, замінюючи будь-яку наявну процедуру з тим самим іменем.

У процесі початкової трансляції також створюється список команд, які утворюють чергу. Команди класифікують на *термінові* та *нетермінові*. До термінових належать лише команди HALT і SHOW. Решта команд – нетермінові. Термінові команди виконуються відразу після того, як вони були передані об'єктові “Процес моделювання”. Решта команд ставляться в чергу після команд, які ще не були виконані.

Є кілька способів передавання оператора моделі процесові моделювання. Можна використати меню Command (Команда), комбінації „гарячих клавіш” і закріплювати за допомогою журналу налаштувань моделі оператор GPSS за функціональною клавішею.

Для посилання об'єктові “Процес моделювання” списку команд треба

вибрати Command ► Custom... (Команда ► Ввести...), набрати список команд і натиснути ОК. Список команд буде відтрансльований і переданий для виконання.

Можна внести список команд, які найчастіше використовуються, до текстового файлу, а потім застосувати команду INCLUDE для передавання списку команд об'єктові "Процес моделювання". Також можна закріпити команду INCLUDE за функціональною клавішею і мати готовий список команд, який буде виконуватися одним натисканням клавіші.

Команди GPSS World відповідно до свого призначення класифікують на дві групи:

- команди визначення об'єктів GPSS;
- команди управління процесом моделювання.

4.3.2. Команди визначення об'єктів GPSS

- B VARIABLE – визначає булеву змінну;
- EQU – присвоює значення змінній користувача;
- FUNCTION – визначає функцію;
- F VARIABLE – визначає змінну;
- INITIAL – ініціалізує або змінює значення логічного ключа, комірки або матриці;
- MATRIX – визначає матрицю комірок зберігання величин;
- QTABLE – визначає статистику черги (Q-таблиці);
- RMULT – встановлює параметри перших семи генераторів випадкових чисел;
- START – встановлює значення лічильника завершень і запускає процес моделювання;
- STORAGE – визначає кількість каналів БКП;
- VARIABLE – визначає дійсну змінну;
- TABLE – визначає таблицю.

4.3.3. Команди управління процесом моделювання

- CLEAR – скидає накопичену статистику до нуля й вилучає з моделі всі транзакти;
- CONDUCT – запускає експеримент;
- CONTINUE – продовжує процес моделювання;
- EXIT – завершує роботу з GPSS World;
- HALT – зупиняє процес моделювання і вилучає всі команди з черги;
- INCLUDE – зчитує і трансльює додатковий файл моделі;
- INTEGRATE – автоматично інтегрує за часом змінну користувача;

- REPORT – встановлює ім'я файла звіту або робить запит на негайний звіт;
- RESET – скидає накопичену статистику до нуля без скидання генераторів випадкових чисел і таймера модельного часу;
- SHOW – обчислює та відображає вираз;
- STEP – встановлює обмежену кількість входжень до блока;
- STOP – встановлює умову зупинки, яка ґрунтується на кількості входжень транзактів до блока.

Розглянемо команди управління процесом моделювання.

4.3.3.1. Команда CLEAR

Під час здійснення експерименту може виникнути необхідність послідовного виконання кількох прогонів, перед кожним з яких незначно змінюється версія моделі. Для здійснення цих змін за один етап моделювання використовується команда CLEAR. Формат запису команди:

CLEAR [A]

Операнд А може бути ON або OFF. За замовчуванням – ON.

Команда CLEAR повертає процес моделювання у початковий стан. Вона скидає всю накопичену статистичну інформацію, вилучає всі транзакти з процесу моделювання і заповнює всі блоки GENERATE першим транзактом. ОКП і БКП стають доступними, встановлюються у стан незайнятості. Вміст усіх блоків стає нульовим. Генератори випадкових чисел не скидаються.

Якщо в команді CLEAR у полі операнда А зазначено OFF, то комірки, логічні ключі та елементи матриць залишаються без змін.

4.3.3.2. Команда CONDUCT

Команда CONDUCT призначена для запуску експерименту. Формат команди:

CONDUCT [A]

Операнд А – виклик процедури Plus-експерименту. Експеримент має бути попередньо зареєстрований в об'єкті “Процес моделювання”. Якщо в об'єкті “Процес моделювання” лише один експеримент без аргументів, то операнд А не використовується.

Під час експерименту є доступ лише до команди HALT (Зупинка), тобто взаємодія користувача з процесом моделювання обмежена. Користувач може лише встановити показ абсолютного модельного часу (View ► Simulation Clock), а для взаємодії необхідно призупинити експеримент.

4.3.3.3. Команда CONTINUE

Команда CONTINUE призначена для відновлення призупиненого процесу

моделювання. Вона має формат:

CONTINUE

Процес моделювання може бути призупинений, якщо виконана умова завершення або надійшла команда HALT, а також у випадку виникнення помилки або нормального завершення. Умови встановлюються командою STOP або STEP.

Якщо процес моделювання призупиняється за умовою завершення, то команда CONTINUE пропускає цю умову, але не вилучає її. Якщо та сама умова буде виконана знову, то процес моделювання знову буде призупинено. Умови завершення можна виключати з моделі командою STOP OFF або явно вилучати у вікні Blocks (Блоки).

Команда CONTINUE може бути використана, коли процес моделювання був перерваний командою HALT. Оскільки команда HALT вилучає всі команди з черги команд, то буде продовжено лише процес моделювання.

Під час здійснення команди CONTINUE спочатку розшукується лічильник завершень. Якщо його значення не дорівнює нулеві, то це означає, що виконання попередньої команди START не було завершено. Тоді викликається планувальник для продовження відпрацювання транзактів. Якщо значення лічильника завершень дорівнює нулеві, то формується стандартний звіт, а планувальник транзактів не викликається.

4.3.3.4. Команда EXIT

Команда EXIT призначена для негайного завершення сеансу роботи з GPSS World. Формат команди:

EXIT [A]

Операнд А – код виходу з GPSS World. Він використовується для управління записом об'єктів “Модель” і „Процес моделювання” у файли. Допустимі значення:

- 1 – всі об'єкти зберігаються;
- -1 – всі об'єкти не зберігаються;
- 0 або за замовчуванням – кожен змінений файл викликає появу вікна із запитом про його збереження.

Для закриття вікна слугує також бібліотечна процедура Exit(). Наприклад, Exit(-1) – всі об'єкти не зберігати.

GPSS World підтримує фоновий пакетний режим роботи, який використовується, коли потрібне тривале моделювання без взаємодії з користувачем. Команду EXIT застосовують у пакетному режимі для завершення сеансу роботи без віконних операцій. Для цього до моделі вводять команду EXIT 1 або виклик процедури Exit(1), щоб завершити сеанс роботи й зберегти всі об'єкти.

4.3.3.5. Команда HALT

Команда HALT призначена для переривання процесу моделювання та вилучення з черги всіх команд. Команда не має операндів:

HALT

Команда HALT є терміною, тому вона не вноситься до черги команд, а виконується негайно. Її виконання зупиняє процес моделювання і вилучає решту команд з черги. Для продовження процесу моделювання потрібно використати команду CONTINUE.

4.3.3.6. Команда INCLUDE

Команда INCLUDE призначена для зчитування і трансляції додаткового файлу моделі з операторами або списком команд. Формат команди:

INCLUDE A

Операнд A – специфікація файлу (повний шлях доступу до файлу). Допустимі значення – рядок. Наприклад:

INCLUDE "Dzad.txt"

INCLUDE "E:\Pryk\Dzad.txt"

У першому прикладі шлях доступу до файлу не наводиться, оскільки припускають, що файл із вказаним іменем є в папці моделі. У другому прикладі зазначено шлях доступу до файлу.

Команда INCLUDE є терміною командою. Під час трансляції вона замінюється файлом. Тому розташовувати в моделі команду INCLUDE треба там, де повинні бути оператори або команди, які містяться у файлі.

Усі файли, які додатково вводяться, нумеруються транслятором цілими числами, починаючи з нуля. Номер 0 присвоюється об'єктові-моделі. Кільком введенням одного файлу також присвоюються унікальні номери, тобто кожне введення файлу призводить до створення наборів блоків, які не є ідентичними.

Команда INCLUDE допускає п'ять рівнів вкладання файлів моделі. Не можна вводити команду INCLUDE в Plus-процедуру.

4.3.3.7. Команда REPORT

Команда REPORT викликає негайне створення стандартного звіту. Формат команди:

REPORT [A],[B]

Операнди A і B є необов'язковими і залишені для сумісності з попередніми версіями GPSS. Тому команди

REPORT

REPORT ,NOW

ідентичні і обидві викликають негайне створення стандартного звіту.

Коли об'єкт „Процес моделювання” натрапляє на команду REPORT, то

негайно починає створюватися стандартний звіт відповідно до параметра In Windows (У вікні), який встановлюється на сторінці Reports у журналі налаштувань (Settings).

Якщо параметр In Windows встановлений, то створюється новий об'єкт-звіт, який містить новий стандартний звіт. Пізніше він може бути збережений або не збережений у файлі. У випадку збереження ім'я файлу має ввести користувач.

Якщо параметр In Windows не встановлений, то створюється новий стандартний звіт, якому присвоюється порядковий номер і він записується у файл.

Зазвичай, немає необхідності використовувати команду REPORT, оскільки стандартний звіт створюється автоматично у вікні Report і пізніше може бути збережений у файлі.

4.3.3.8. Команда RESET

Якщо моделювання виконується довго, то в системі може досягатися стаціонарний стан. Скільки часу треба здійснювати моделювання, щоб досягти стаціонарного стану? Часто відповідь на це питання можна отримати з досвіду експериментування з моделлю. Зокрема, для цього слугує команда RESET, яка скидає статистику до нуля, не здійснюючи вилучення транзактів з процесу моделювання. Команда RESET не має операндів:

RESET

Команда RESET не скидає генератори випадкових чисел, не впливає на абсолютний модельний час і нумерацію транзактів, проте скидає до нуля відносний модельний час (час після останньої команди RESET). Отже, за допомогою оператора RESET забезпечується зручний засіб збирання статистичних даних у типових умовах функціонування систем.

4.3.3.9. Команда SHOW

Команда SHOW призначена для передавання об'єктові "Процес моделювання" виразу для обчислення та запису результату в рядок стану вікна Model. Формат команди:

SHOW X

X – арифметичний або логічний вираз. Команда виконується терміново. Якщо у виразі є СЧА, то вони обчислюються для активного транзакта. У випадку відсутності активного транзакта, тобто коли процес моделювання ще не запущено, видається повідомлення про помилку. Змінним користувача, якщо вони застосовуються у виразі, повинні бути попередньо задані значення за допомогою команди EQU.

Наприклад:

SHOW XN1

```
SHOW N$Mit1/N$Mit2  
SHOW LOG(AC1)
```

У першому прикладі в рядку стану показується номер активного транзакта, У другому – результат ділення кількості транзактів, які увійшли до блока з міткою Mit1, на кількість транзактів, які увійшли до блока з міткою Mit2. У третьому прикладі обчислюється і виводиться в рядок стану натуральний логарифм абсолютного модельного часу.

4.3.3.10. Команда STEP

Команда STEP призначена для організації виконання процесом моделювання заданої кількості входжень транзактів до блоків. Формат команди:

```
STEP [A]
```

Операнд А – кількість входжень транзактів до блоків. Наприклад, команду

```
STEP 1
```

використовують для покрокового виконання процесу моделювання, тобто процес моделювання призупиняється після одного входження транзакта до блока. Одночасно з цим у вікні JOURNAL з'являється повідомлення, в якому зазначено час, номер активного транзакта, поточний блок, у якому перебуває активний транзакт, і наступний блок, до якого має увійти транзакт.

Команду STEP зручно використовувати для тестування (відлагодження) процесу моделювання. Покрокове виконання процесу моделювання, крім вікна JOURNAL, динамічно відображається у вікні Blocks (Блоки), що дає змогу простежити за шляхами руху транзактів і за логікою роботи моделі. Команду можна вводити натисканням комбінації [Ctrl]+[Alt]+[1] або за допомогою миші у вікні Blocks.

4.3.3.11. Команда STOP

Команда STOP призначена для встановлення або знімання умови переривання процесу моделювання. Формат команди:

```
STOP [A],[B],[C]
```

Значення операндів: А – номер транзакта; В – номер блока; С – визначає стан команди, може бути ON або OFF, за замовчуванням – ON.

Наприклад:

```
STOP 28,5
```

Така команда вводить умову, яка призведе до зупинення процесу моделювання в момент, коли транзакт номер 28 намагатиметься увійти до блока номер 5. У рядок стану і у всі відкриті вікна JOURNAL передається час, номер активного транзакта, поточний блок, у якому перебуває активний транзакт, і наступний для його входження блок.

Для продовження моделювання використовують команду CONTINUE.

Вона не знімає умови зупинення. Якщо ця умова знову виконається, то процес моделювання знову зупиниться. Для зняття всіх умов зупинення потрібно використати команду

```
STOP  ,,OFF
```

Наведемо ще приклади команд:

```
STOP  25
```

```
STOP  25,OFF
```

```
STOP  ,Mit3
```

```
STOP  ,Mit3,OFF
```

Перша команда зупиняє процес моделювання, коли транзакт номер 25 стає активним. Друга команда знімає умову, встановлену першою командою. Третя призводить до зупинки процесу моделювання, коли будь-який транзакт намагається увійти до блока з міткою Mit3. Четверта команда знімає цю умову.

За допомогою команди STOP на етапі тестування і верифікації можна простежити правильність виконання процесу моделювання. Команду вводять за допомогою миші у вікні Blocks.

4.4. Вікна GPSS World

Для спостереження за процесом моделювання і дією на нього команд на етапі тестування і верифікації використовують десять графічних вікон. Динамічні вікна (Window ► Simulation Window) класифікують за типами об'єктів:

- блоки (Blocks);
- вирази (Expression);
- одноканальні пристрої (Facilities);
- логічні ключі (Logicswitches);
- матриця (Matrix);
- графік (Plot);
- черги (Queues);
- комірки (Savevalues);
- багатоканальні пристрої (пам'яті) (Storages);
- таблиці (Table).

Вікно Blocks показує входження транзактів до блоків. Воно дає змогу за допомогою миші або клавіатури встановлювати і вилучати контрольні точки і візуально відстежувати пересування транзактів.

Вікно Expression призначене для спостереження за змінами будь-якої кількості Plus-виразів.

Вікно Plot дає змогу одночасно спостерігати за будь-якою кількістю кольорових графіків. Ці графіки можна зберегти або роздрукувати. Кожне вікно виводить від одного до восьми графіків змінних, у тім числі інтегрованих. Масштаб графіків можна змінювати.

Вікно Table є динамічною гістограмою, корисною для спостереження за збиранням даних і оцінки збіжності до породжуючого ймовірнісного розподілу.

Вікна Blocks, Facilities, Logicswitches, Queues, Savevalues, Storages мають детальний огляд і загальний огляд. Відкриваються вони завжди з детальним оглядом. Перехід до загального огляду здійснюють за допомогою View ► Entity Details (Вигляд ► Подобиці).

Крім динамічних вікон, у GPSS World доступні (Window ► Simulation Snapshot) кадри стану для:

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

Динамічні вікна обновлюються оперативно з метою відповідності до зміни стану процесу моделювання. На відміну від динамічних вікон кадри стану – це статичні відображення об'єктів, зафіксовані в певний момент часу

виконаного процесу моделювання, тобто кадри стану динамічно не обновлюються.

Опишемо застосування розглянутих засобів GPSS World для відлагодження моделей.

4.5. Відлагодження моделей

Відкриємо модель прикладу 3.7 (File►Open, потім знайдемо файл Prykl 3.7.gps і натиснемо Open). Відкриємо журнал налаштувань моделі Edit►Settings і виберемо сторінку Function Keys (Функціональні клавіші) (рис. 4.12). У полі клавіші [F8] наберемо команду:

SHOW Q1

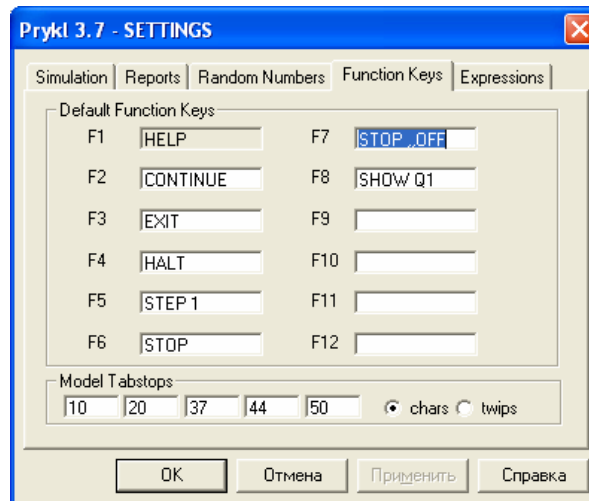


Рис. 4.12. Сторінка „Function Keys” вікна журналу налаштувань

Натиснемо „Применить” і ОК. Тепер команда закріплена за функціональною клавішею, і під час виконання процесу моделювання можна інтерактивно досліджувати поточну довжину черги номер 1 (черги БКП System) натисканням лише однієї клавіші [F8].

Запустимо процес моделювання, вибравши Command► Create Simulation. Відкриємо також вікно Blocks. Для цього в головному меню виберемо Window ► Simulation Window ► Blocks Window (рис. 4.13).

Поставимо умову зупинки на активний транзакт номер 40. Для цього виберемо Command ► Custom... і в діалоговому вікні (рис. 4.14) наберемо STOP 40 і натиснемо ОК.

У головному меню виберемо Command ► START 1, натиснемо ОК. З’явиться повідомлення, що у процесі моделювання транзакт номер 40 зупинився під час першої спроби увійти до блока. У цьому випадку це блок GENERATE.

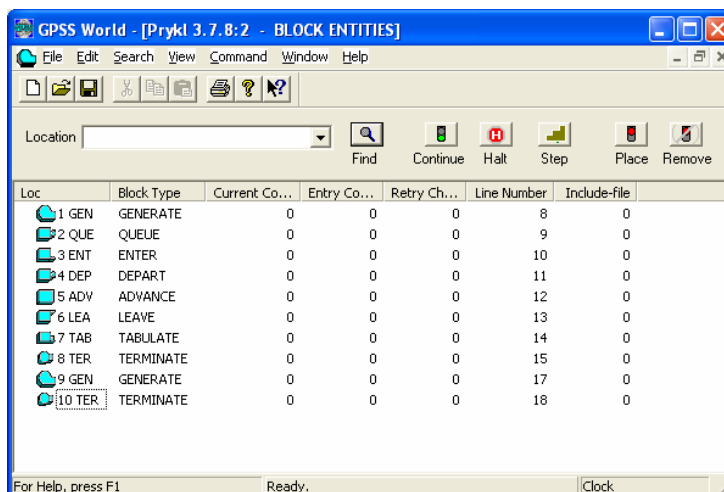


Рис. 4.13. Вікно Blocks моделі Prykl. 3.7

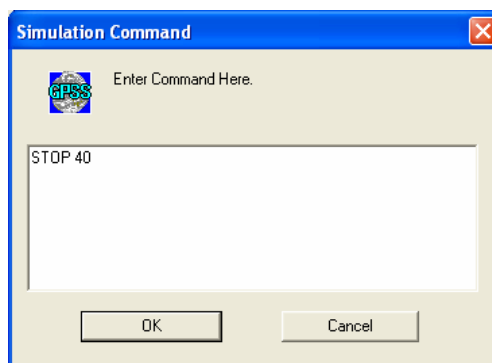


Рис. 4.14. Вікно Command ► Custom...

Тепер почнемо виконувати процес моделювання в покроковому режимі, використовуючи функціональну клавішу [F5] із закріпленою за нею в журналі налаштувань моделі командою STEP 1 (див. рис. 4.12). Спочатку знімемо умову зупинки. Для цього виберемо Window ► Simulation Snapshot ► User Stops (Умови зупинки), клацнемо мишею на 40, натиснемо Remove (Вилучити) і ОК.

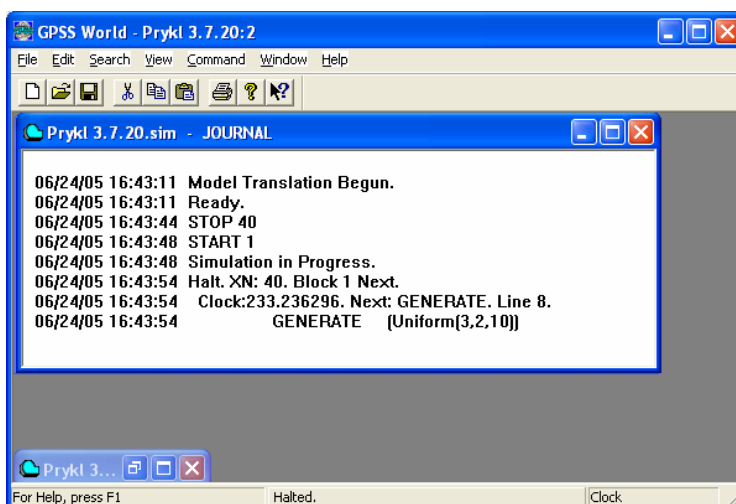


Рис. 4.15. Повідомлення про зупинку у вікні Journal

Виберемо команду STEP 1, натиснувши функціональну клавішу [F5]. Можна було також виконати цю команду натисканням комбінації клавіш [Ctrl]+[Alt]+[S] або вибравши Command ► STEP в головному меню. Транзакт номер 40 увійде до блока GENERATE. Це можна побачити в біжучому повідомленні вікна Journal.

Дослідимо деякі показники процесу моделювання. Виберемо Command ► SHOW... і в діалоговому вікні (рис. 4.16) наберемо Q1, натиснемо OK.

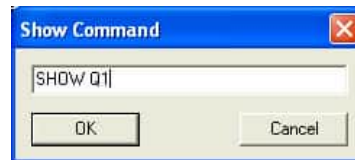


Рис. 4.16. Діалогове вікно Command ► SHOW...

Довжина (0) черги номер 1 відображається у рядку стану головного вікна і у вікні Journal.

Виберемо Window ► Simulation Window ► Queues Window. Відкриється вікно (рис. 4.17) з детальним оглядом стану черги. Для спостереження за зміною параметрів черги натиснемо кілька разів [F5]. Закриємо вікно.

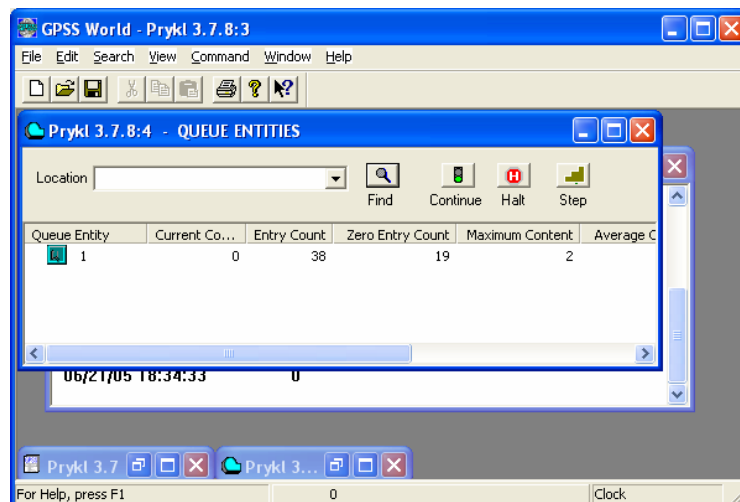


Рис. 4.17. Вікно Queues (Черги). Детальний огляд

Виберемо Window ► Simulation Window ► Storages Window. Відкриється вікно (рис. 4.18) з детальним оглядом БКП. Можна поспостерігати за зміною стану пристрою, натиснувши клавішу [F2] (CONTINUE), а потім [F4] (HALT).

Для переходу до загального огляду вікна Storages виберемо View ► Entity Details. Відкриється вікно (рис. 4.19). Натиснемо функціональну клавішу [F2] (CONTINUE), а потім [F4] (HALT). Звернемо увагу на зміну стану пристрою System. Колір значка пристрою змінюється із сірого, коли він вільний, на червоний, коли пристрій зайнятий. Маленька біла смужка у правій частині

значка показує невелику чергу. Велика біла смужка означає, що черга перевищує десять транзактів. Закриємо вікно Storages.

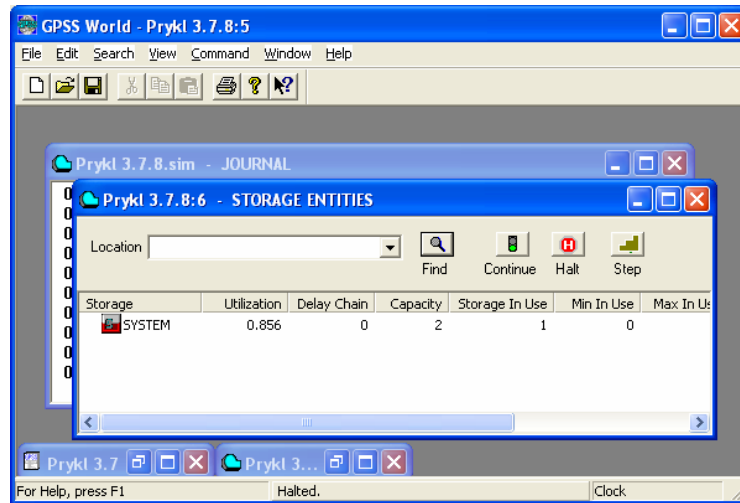


Рис. 4.18. Вікно Storages (БКП). Детальний огляд

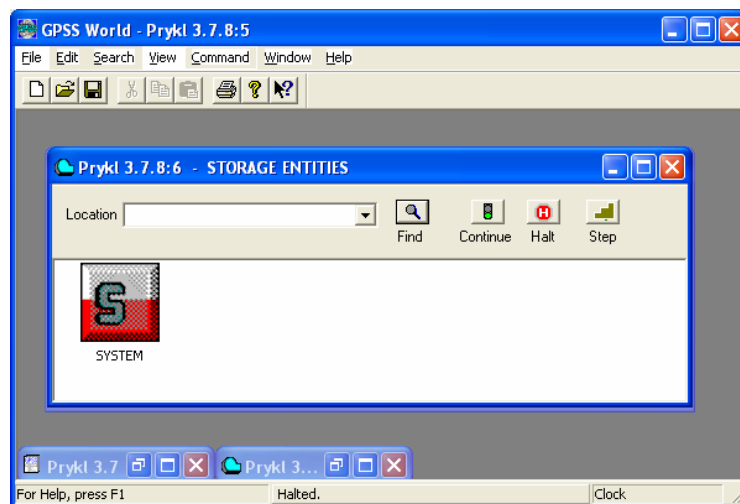


Рис. 4.19. Вікно Storages (БКП). Загальний огляд

Відкриємо вікно Expression (Вирази). Виберемо Window ► Simulation Window ► Expression Window. З'явиться вікно (рис. 4.20).

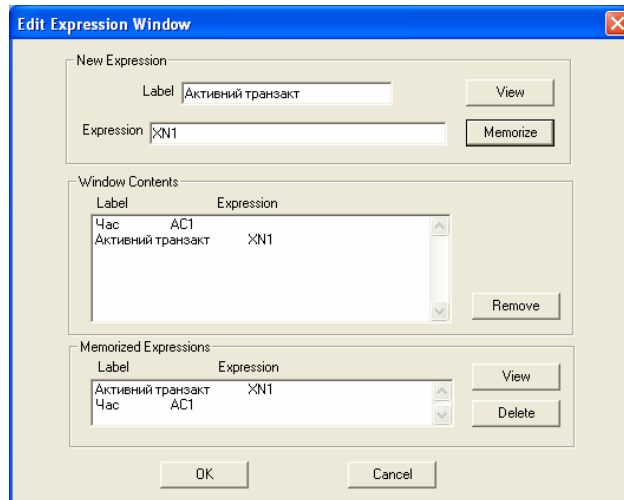


Рис. 4.20. Вікно Edit Expression з введеними виразами та їхніми мітками

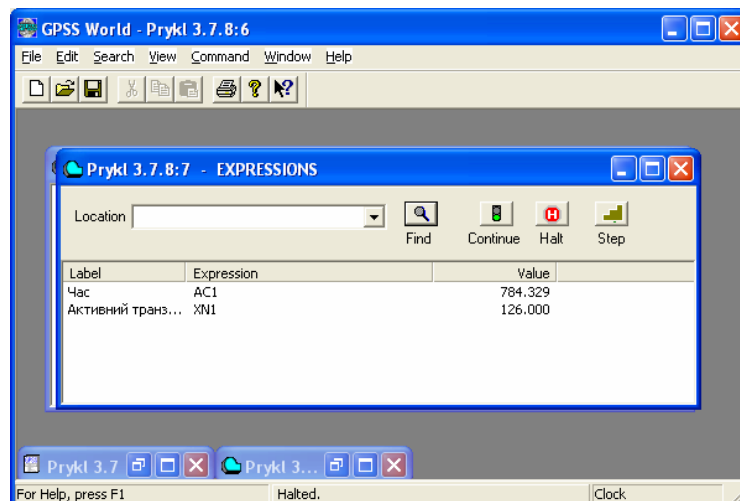


Рис. 4.21. Вікно Expression

У діалоговому вікні потрібно ввести мітку та вираз, який нам треба обчислити. У полі Label (Мітка) введемо:

Час

У полі Expression наберемо:

AC1.

Натиснемо кнопку View (Перегляд) і Memorize (Запам'ятати). Це потрібно для того, щоб показати значення цього виразу в поточному процесі моделювання та переглядати цей вираз після збереження процесу моделювання і повторного його запуску.

Тепер у діалогове поле Label замість **Час** введемо:

Активний транзакт

а замість **AC1** введемо:

XN1

Знову натиснемо кнопки View і Memorize. Ці кнопки потрібно натискати після кожного введення мітки і виразу. Натиснемо ОК. Ми побачимо (рис. 4.21) абсолютний модельний час і номер активного транзакта.

Простежимо за зміною значень виразів, натискаючи клавішу [F2] (CONTINUE), а потім [F4] (HALT).

Тепер закриємо вікно Expression і знову відкриємо. У відкритому вікні Edit Expression (див. рис. 4.20) вирази та їхні мітки збереглися лише у списку Memorized Expression (Збережені вирази). Можна вибрати вирази, які необхідно відобразити. Для цього потрібно вибрати вираз у списку Memorized Expression і натиснути кнопку View. Виберемо Час і натиснемо кнопку View, потім – Активний транзакт і знову натиснемо кнопку View. Обидва вирази з’являться у полі Window Contents. Натиснемо ОК. Ми побачимо вікно Expression, яке показує абсолютний модельний час і номер активного транзакта. Закриємо вікно.

Наголосимо, що для переходу від поля до поля у вікні Edit Expression не можна використовувати клавішу [Enter], оскільки це спричинить закриття діалогового вікна і видасть помилку про недопустимий вираз, якщо не введені дані до кінця. Для переходу можна користуватися мишею або клавішею [Tab].

Завершимо процес моделювання, закриємо всі вікна, крім вікна з об’єктом “Модель”. Знову виберемо Command► Create Simulation.

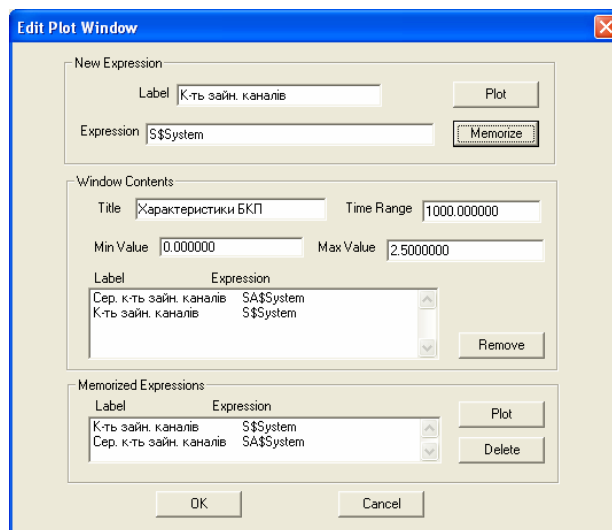


Рис. 4.22. Діалогове вікно Edit Plot з введеними даними

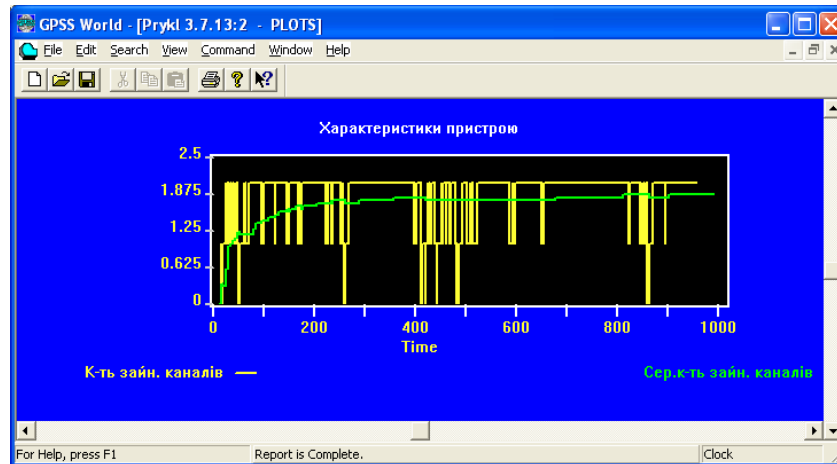


Рис. 4.23. Вікно Plot (Графік) з двома побудованими графіками

Виберемо Window ► Simulation Window ► Plot Window. З'явиться вікно Edit Plot (Редактор графіка) (рис. 4.22). Воно схоже на вікно Edit Expression, але на відміну від нього призначене для задання осей, виразів та інших загальних даних, які вводять у графік.

Поле Label повинне містити пояснення до кривої графіка. У це поле введемо:

Сер. к-ть зайн. каналів

У поле Expression введемо першу змінну для побудови графіка:

SA\$System

СЧА SA описує середню кількість зайнятих каналів.

У полі Title (Заголовок) задамо загальне ім'я всіх кривих, які будуть зображені на графіку:

Характеристики пристрою

Для кожного графіка має бути зазначений інтервал часу. У полі Time Rang (Інтервал часу) введемо 1000 – максимальне значення на осі часу (осі абсцис). Для осі ординат введемо значення від 0 до 2,5. Отриманий проміжок [0;2,5] міститиме можливі значення 0; 1; 2 змінної SA для двоканального пристрою. Натиснемо Plot (Побудувати) і Memorize.

Введемо змінну для другого графіка. У полі Label наберемо:

К-ть зайн. каналів

У полі Expression введемо:

S\$System

СЧА S визначає кількість зайнятих каналів. Знову натиснемо Plot, Memorize і OK. Ми побачимо порожнє (без побудованих графіків) вікно Plot. Тепер запусимо процес моделювання. Після завершення виконання моделі вікно Plot виглядатиме так, як показано на рис. 4.23.

У одному вікні можна будувати до восьми графіків. Для цього потрібно ввести відповідну кількість міток і виразів.

Закриємо вікно Plot. Виберемо File► Save (Файл► Зберегти). Якщо раніше

файл не зберігали, то за замовчуванням буде задано ім'я Pрыkl 3.7.1.sim. Натиснемо Save і закриємо всі відкриті вікна.

Продовжимо збережений процес моделювання. Виберемо File► Open (Файл► Відкрити). У випадаючому списку в нижній частині вікна виберемо тип файла Simulation (Процес моделювання), потім у верхній частині діалогового вікна виберемо Pрыkl 3.7.1.sim і натиснемо Open.

Відкрився збережений файл процесу моделювання. Виберемо Window ► Simulation Window ► Plot Window. З'явиться вікно Edit Plot. У списку Memorized виділимо першу мітку і натиснемо Plot, потім другу і також натиснемо Plot. Залишилося ввести в полі Title заголовок графіка, у полі Time Range інтервал часу – 1000 і в полі Max Value (Максимальне значення) – 2.0 або 2.5. Тепер все готове для побудови графіків на проміжку часу (1000,2000). Для цього досить знову запустити процес моделювання командою START 1.

Закриємо всі вікна і знову відкріємо файл Pрыkl 3.7.gps. Запустимо процес моделювання, вибравши Command► Create Simulation. Виберемо Window► Simulation Window ► Table Window. У вікні Open Table (рис. 4.24) виберемо TTIME і натиснемо OK.

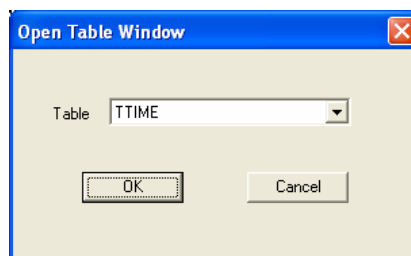


Рис. 4.24. Діалогове віно Open Table (Вибрати таблицю)

Ми побачимо відображення таблиці TTIME, але крім осей, на гістограмі поки що нічого не буде. Запустимо процес моделювання командою START 1. Після завершення моделювання отримаємо гістограму розподілу часу перебування замовлення у системі (рис. 4.25), на якій видно, що середній час становить 16,681 з середнім квадратичним відхиленням 8,904.

Закриємо всі вікна і знову створимо процес моделювання. Потім виберемо Window► Simulation Window ► Blocks Window. У вікні Blocks (Блоки) виберемо блок ADVANCE (блок 5). Це можна зробити, перемістивши курсор миші до блока 5 і натиснувши ліву кнопку миші. Клацнемо на блоці ADVANCE, потім натиснемо на кнопку Place (Розміщувати) на панелі інструментів вікна Blocks. Тим самим у блоці 5 визначена умова зупинки.

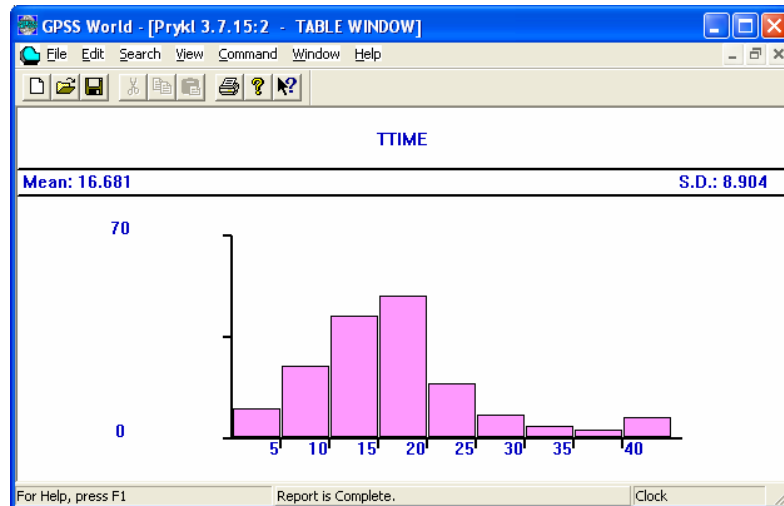


Рис. 4.25. Вікно Table з результатами моделювання

Запустимо процес моделювання. Виберемо Command ► START 1, натиснемо OK. Процес моделювання зупиниться, коли перший транзакт намагатиметься потрапити до блока ADVANCE.

Вилучимо умову зупинки з блока ADVANCE. Виберемо Window ► Simulation Snapshot ► User Stops (Умови зупинки), натиснемо кнопку Remove All (Вилучити все) (рис. 4.26) і кнопку OK.

В GPSS World передбачено можливості задання різного розташування вікон, які з'являються в головному вікні під час роботи з моделлю. Якщо вибрати Window ► Cascade (Послідовно), то вікна розташуються так, як показано на рис. 4.27. Тепер виберемо Window ► Tile (Мозаїка). Отримаємо розташування вікон одного під одним (рис. 4.28).

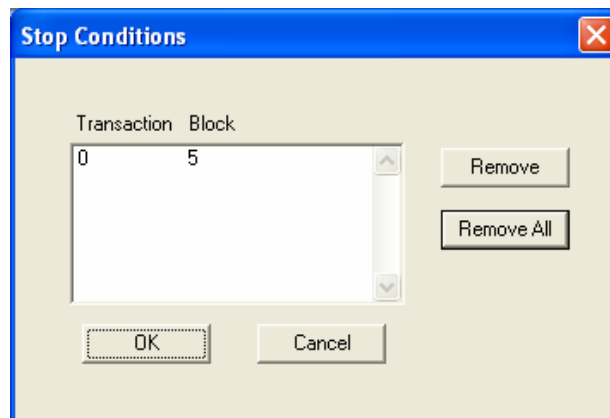


Рис. 4.26. Вікно Stop Conditions (Умови зупинки)

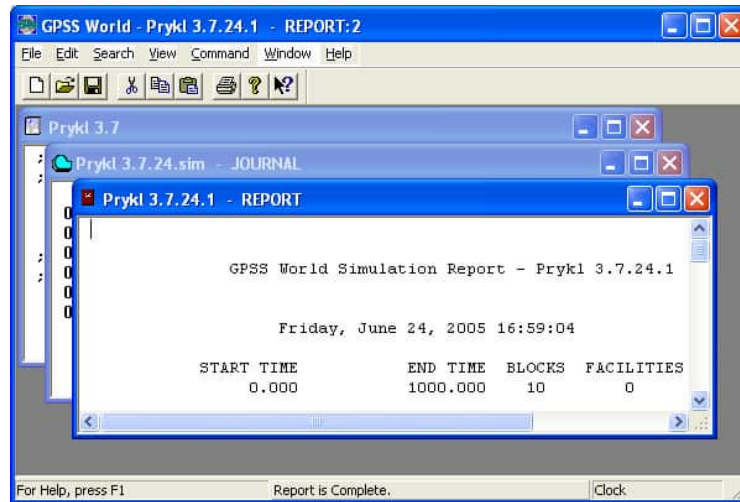


Рис. 4.27. Послідовне розташування вікон у головному вікні

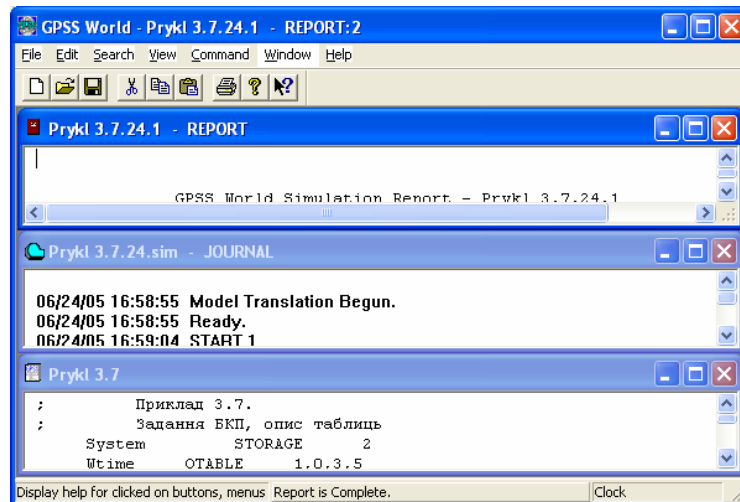


Рис. 4.28. Мозаїчне розташування вікон у головному вікні

4.6. Внесення змін до моделі в інтерактивному режимі

Розглянемо приклади застосування команд GPSS World для внесення змін до створеного після початкової трансляції об'єкта „Процес моделювання”.

Внесемо незначні зміни до моделі прикладу 3.7 для полегшення роботи з нею в інтерактивному режимі.

Приклад 4.1. *Модель прикладу 3.7, пристосована до роботи в інтерактивному режимі:*

```
; Приклад 4.1.  
; Визначення об'єктів GPSS  
Inter VARIABLE (Uniform(3,2,10))  
Zatr VARIABLE (Uniform(5,2,20))  
System STORAGE 2  
Wtime QTABLE 1,0,3,5  
Ttime TABLE M1,5,5,9  
; Сегмент імітування надходження та обслуговування  
; замовлень і збору статистики очікування  
GENERATE V$Inter  
QUEUE 1  
ENTER System  
DEPART 1  
ADVANCE V$Zatr  
LEAVE System  
TABULATE Ttime  
TERMINATE  
; Сегмент задання часу моделювання  
GENERATE 1000  
TERMINATE 1
```

У моделі введено арифметичні змінні Inter і Zatr, які дають нам змогу в інтерактивному режимі змінювати відповідно інтервал часу між надходженнями транзактів (блок GENERATE) та час обслуговування (блок ADVANCE).

Відкриємо модель прикладу 4.1 (File►Open, потім знайдемо файл Pрыkl 4.1.gps і натиснемо Open). Запустимо процес моделювання, вибравши Command► Create Simulation.

Покажемо, як можна використати команду INCLUDE, наприклад, для інтерактивної зміни кількості каналів БКП System. Виберемо File► New, в діалоговому вікні New клацнемо лівою клавiшею миші на полі Text File і натиснемо ОК. У вікні текстового файлу наберемо:

```
System STORAGE 3
```

тим самим встановивши кількість каналів пристрою System, яка дорівнює трьом. Збережемо текстовий файл під іменем Pr.txt.

Виберемо Command► Custom... і у вікні Simulation Command введемо:

INCLUDE "Pr.txt"

Натиснемо ОК. Запустимо процес моделювання командою **START 1**.

Після завершення моделювання у вікні **TABLE** отримуємо гістограму розподілу часу перебування замовлення у системі (рис. 4.29), яка показує, що середній час становить 11,032 з середнім квадратичним відхиленням 5,117. Порівнюючи ці дані з результатами моделювання для двоканальної СМО (рис. 4.25), бачимо, що зі збільшенням кількості каналів середній час перебування замовлення у системі зменшується.

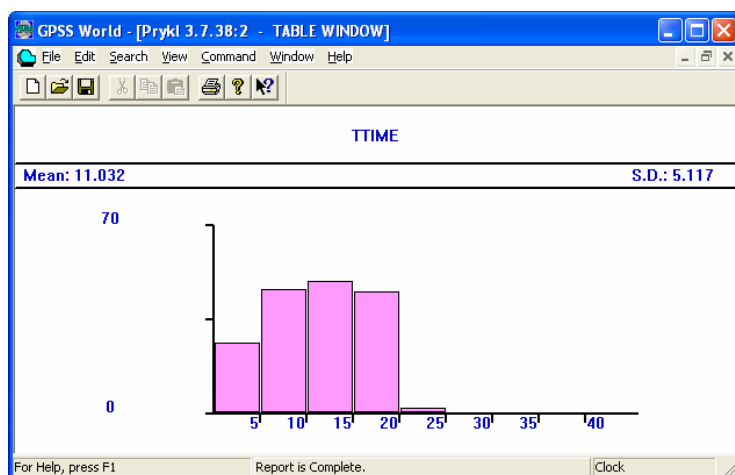


Рис. 4.29. Вікно Table з результатами моделювання після інтерактивної зміни кількості каналів БКП

Однак процес інтерактивної зміни кількості каналів БКП System можна значно спростити шляхом безпосереднього введення до моделі команди **STORAGE**. Для цього після початкової трансляції достатньо вибрати **Command► Custom...**, у вікні **Simulation Command** ввести:

System STORAGE 3

і натиснути ОК.

Тепер після початкової трансляції для СМО з двома каналами змінимо розподіл інтервалу часу між надходженнями транзактів. Виберемо **Command► Custom...**, і у вікні **Simulation Command** введемо:

Inter VARIABLE (Exponential(7,0,12))

збільшивши у два рази середній інтервал надходження транзактів. Натиснувши ОК, запустимо процес моделювання командою **START 1**.

Після завершення моделювання у вікні **TABLE** знову розглянемо гістограму розподілу часу перебування замовлення в системі (рис. 4.30). Вона показує, що середній час становить 13,206 із середнім квадратичним відхиленням 7,448. Порівнюючи ці дані з результатами моделювання прикладу 3.7 (там середній інтервал надходження дорівнює 6 одиниць модельного часу, розподіл рівномірний) (рис. 4.25), доходимо висновку, що зі зменшенням середнього інтервалу надходження замовлень середній час перебування

замовлення в системі також зменшується.

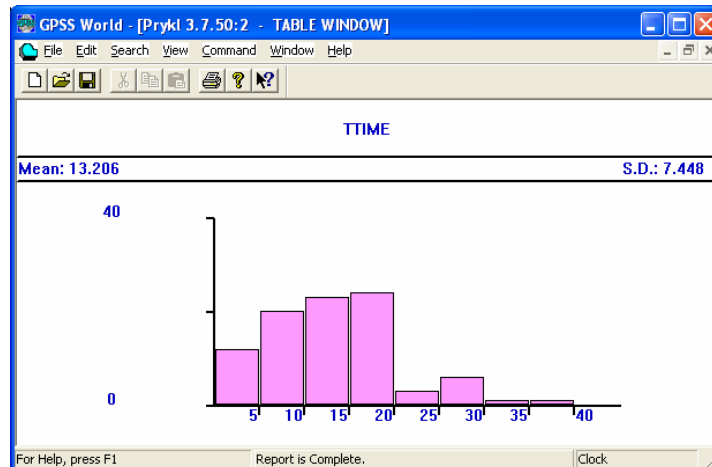


Рис. 4.30. Вікно Table з результатами моделювання після інтерактивної зміни операнда A у блоці GENERATE

Аналогічно можна змінити розподіл часу обслуговування, внісши зміни до операнда A блоку ADVANCE за допомогою арифметичної змінної Zatr. Виберемо Command► Custom..., а у вікні Simulation Command, наприклад, введемо:

```
Zatr VARIABLE (Exponential(32,0,22))
```

Ми збільшуємо у два рази середній час затримки транзактів у блоці ADVANCE, а розподіл часу затримки змінюємо з рівномірного на показниковий. Після завершення моделювання у вікні TABLE прочитаємо: середній час перебування замовлення в системі у цьому випадку становить 220,56, а середнє квадратичне відхилення – 138,399. Ці дані також можна знайти безпосередньо у стандартному звіті.

4.7. Елементи стандартного звіту

Після завершення роботи моделі система автоматично створює звіт. Статистична інформація виводиться лише про ті об'єкти, які є в моделі. У загальному випадку стандартний звіт містить такі елементи.

Рядок заголовка

GPSS World Simulation Report - Pрыkl 3.7.85.1

Saturday, June 25, 2005 16:52:12

Містить ім'я файлу моделі, за результатами прогону якої створено звіт, дату і час прогону моделі.

Загальна інформація про результати роботи моделі

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1000.000	10	0	1

- START TIME – початковий час, тобто абсолютний модельний час у момент початку моделювання. Він дорівнює абсолютному модельному часові завдяки дії оперетора RESET або CLEAR;
- END TIME – кінцевий час, тобто значення абсолютного модельного часу в момент, коли лічильник завершень набуває значення 0;
- BLOCKS – кількість блоків, використаних у моделі в момент завершення моделювання;
- FACILITIES – кількість одноканальних пристроїв, використаних у моделі в момент завершення моделювання;
- STORAGES – кількість багатоканальних пристроїв, використаних у моделі в момент завершення моделювання.

Інформація про імена

NAME	VALUE
INTER	10000.000
SYSTEM	10002.000
TTIME	10004.000
WTIME	10003.000
ZATR	10001.000

- NAME – перелік заданих користувачем імен, які містяться у програмі моделі;
- VALUE – числове значення, яке присвоюється імені. Система починає відлік з 10000.

Інформація про блоки

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1	GENERATE		169	0	0
2	QUEUE		169	0	0

3	ENTER	169	0	0
4	DEPART	169	0	0
5	ADVANCE	169	2	0
6	LEAVE	167	0	0
7	TABULATE	167	0	0
8	TERMINATE	167	0	0
9	GENERATE	1	0	0
10	TERMINATE	1	0	0

- LABEL – мітка, тобто алфавітно-цифрове ім'я цього блока (якщо воно задане);
- LOC – числовий номер позиції цього блока в моделі;
- BLOCK TYPE – тип блока GPSS;
- ENTRY COUNT – кількість транзактів, які увійшли до цього блока з початку роботи програми чи після останнього виконання команди RESET або CLEAR;
- CURRENT COUNT – кількість транзактів, які перебувають у даному блоці в момент завершення моделювання;
- RETRY COUNT – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього блока.

Інформація про одноканальні пристрої

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER
SYSTEM 94	0.993	10.564	1	95	0	0	75

- FACILITY – ім'я або номер пристрою;
- ENTRIES – кількість разів, коли пристрій був зайнятий або зайнятий з перериванням з початку моделювання чи після останнього виконання команди RESET або CLEAR;
- UTIL. – коефіцієнт використання пристрою, тобто частка часу моделювання, протягом якого пристрій був зайнятим;
- AVE. TIME – середній час займання пристрою одним транзактом з початку моделювання чи після останнього виконання команди RESET або CLEAR;
- AVAIL. – стан пристрою в кінці моделювання (1, якщо пристрій доступний; 0 – якщо недоступний);
- OWNER – номер транзакта, який займає пристрій (0 – пристрій не зайнятий);
- PEND – кількість транзактів, які чекають виконання з перериванням інших транзактів (тобто тих, які ввійшли до блока PREEMPT у режимі переривання);
- INTER – кількість транзактів, перерваних на даний момент (кількість транзактів у списку переривань);
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка

залежить від стану цього пристрою;

- DELAY – кількість транзактів, які чекають займання пристрою (сюди входять також транзакти, які чекають займання пристрою в режимі переривання за допомогою блоків PREEMPT).

Інформація про об'єкти типу “черга”

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
1	6	0	169	45	0.983	5.819	7.931	0

- QUEUE – ім'я або номер черги;
- MAX – максимальна довжина черги протягом періоду моделювання, який починається з моменту трансляції моделі чи застосування команди RESET або CLEAR;
- CONT. – поточна довжина черги в кінці процесу моделювання;
- ENTRY – загальна кількість входжень транзактів у чергу протягом часу моделювання;
- ENTRY(0) – загальна кількість входжень транзактів у чергу з нульовим часом очікування;
- AVE.CONT. – середня довжина черги протягом часу моделювання;
- AVE.TIME – середній час перебування одного транзакта в черзі з урахуванням усіх входжень до черги;
- AVE.(-0) – середній час перебування одного транзакта в черзі без урахування „нульових” входжень до черги;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану черги.

Інформація про об'єкти типу “багатоканальний пристрій”

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
SYST	2	0	0	2	169	1	1.815	0.907	0	0

- STORAGE – ім'я або номер пристрою;
- CAP. – ємність пам'яті (кількість каналів), задана оператором STORAGE;
- MIN. – мінімальна кількість одиниць пам'яті (кількість каналів), які були використані за час моделювання;
- MAX. – максимальна кількість одиниць пам'яті (кількість каналів), які були використані за час моделювання;
- ENTRIES – кількість входжень до пристрою за період моделювання;
- AVL. – стан пристрою в кінці моделювання (1, якщо пристрій доступний; 0 – якщо недоступний);
- AVE.C. – середня кількість зайнятих каналів за період моделювання;
- UTIL. – коефіцієнт використання пристрою;

- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього пристрою;
- DELAY – кількість транзактів, які чекають у блоках ENTER, пов'язаних з цим пристроєм.

Інформація про таблиці

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
WTIME	5.819	7.706		0		
			- 0.000		45	26.63
			0.000 - 3.000		36	47.93
			3.000 - 6.000		30	65.68
			6.000 - 9.000		24	79.88
			9.000 -		34	100.00

- TABLE – ім'я або номер таблиці або Q-таблиці;
- MEAN – середнє значення табульованого аргумента;
- STD.DEV. – середнє квадратичне відхилення табульованого аргумента;
- RANGE – нижня і верхня межі частотного класу; частотні класи, сумарне значення яких дорівнює 0, у файл статистики не виводяться;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цієї таблиці;
- FREQUENCY – сумарна величина (частота), яка формується під час потрапляння табульованого аргумента у зазначені межі (значення операнда В сумуються блоком TABULATE);
- CUM.% – величина частоти у відсотках до загальної кількості значень табульованого аргумента.

Інформація про списки користувача

USER CHAIN	SIZE	RETRY	AVE.CONT	ENTRIES	MAX	AVE.TIME
BUF	0	0	0.983	124	6	7.931

- USER CHAIN – ім'я або номер списку користувача;
- SIZE – кількість транзактів у списку користувача в кінці часу моделювання;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього списку користувача;
- AVE.CONT. – середній вміст списку користувача протягом часу моделювання;
- ENTRIES – кількість входжень до списку користувача протягом часу моделювання;
- MAX – максимальна кількість транзактів у списку користувача за період моделювання;

- AVE.TIME – середній час перебування транзакта у списку користувача.

Інформація про групи транзактів

XACT GROUP	GROUP SIZE	RETRY
USERGR	7	0

- XACT GROUP – ім'я або номер групи транзактів;
- GROUP SIZE – кількість транзактів у групі в кінці моделювання;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану даної групи транзактів.

Інформація про числові групи

NUMERIC GROUP	GROUP SIZE	RETRY
USNUM	2	0

- NUMERIC GROUP – ім'я або номер числової групи;
- GROUP SIZE – кількість транзактів, які містяться в числовій групі в кінці моделювання;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану даної числової групи.

Інформація про логічні перемикачі

LOGICSWITCH	VALUE	RETRY
KLUCH1	1	0

- LOGICSWITCH – ім'я або номер логічного ключа;
- VALUE – значення ключа в кінці моделювання (1 (true) – „увімкнений”; 0 (false) – „вимкнений”);
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану цього логічного ключа.

Інформація про збережені величини (комірки зберігання)

SAVEVALUE	RETRY	VALUE
IMO	0	0.585

- SAVEVALUE – ім'я або номер комірки;
- VALUE – значення збереженої величини в кінці моделювання;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану даної комірки.

Інформація про матриці

MATRIX	RETRY	INDICES	VALUE
PLAN	0		
		1 1	34
		1 2	27
		1 3	18

- MATRIX ім'я або номер матриці;
- RETRY – кількість транзактів, які чекають виконання спеціальної умови, яка залежить від стану даної матриці;
- INDICES – до шести цілих чисел, які визначають елементи матриці;
- VALUE – значення елемента матриці в кінці моделювання (елементи, які дорівнюють нулю, виводяться у звіті групами).

Інформація про списки поточних і майбутніх подій

CEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
2	0	0.000	2	26	27		
						KPOVT	0.000

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
169	0	1001.091	169	5	6		
170	0	1001.898	170	5	6		
171	0	1005.087	171	0	1		
172	0	2000.000	172	0	9		

Елементи статистики для списку поточних подій (CEC) мають такий зміст:

- CEC XN – номер транзакта, який міститься у списку поточних подій;
- PRI – пріоритет транзакта;
- M1 – час входження транзакта до моделі або час першого транзакта, який його породив;
- ASSEM – номер сім'ї цього транзакта;
- CURRENT – номер блока, в якому перебуває транзакт у кінці моделювання;
- NEXT – номер наступного блока, до якого мав увійти транзакт;
- PARAMETER – ім'я або номер параметра транзакта;
- VALUE – значення параметра.

У змісті статистики для списку майбутніх подій (FEC) є лише одна відмінність. Замість поля M1 використовується поле BDT, яке визначає момент абсолютного модельного часу, коли транзакт покине список майбутніх подій.

5. МОДЕЛІ РОЗІМКНЕНИХ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

5.1. Класичні системи: порівняння з аналітичними моделями

Для більшості розімкнених систем масового обслуговування аналітичні моделі побудовано лише у випадку показникового розподілу інтервалу надходження замовлень і часу обслуговування. За таких умов можна здійснити порівняння результатів, отриманих за допомогою аналітичних та імітаційних моделей цих систем.

Використання аналітичної моделі СМО призводить до необхідності інтегрування системи звичайних диференціальних рівнянь для ймовірностей станів системи. Отримання аналітичних розв'язків тут можливе лише у випадку, коли кількість рівнянь системи не перевищує трьох. Числове інтегрування системи звичайних диференціальних рівнянь можна здійснити за допомогою системи GPSS World.

В GPSS World інтегрування виконується модифікованим методом Рунге-Кутта-Фехлберга п'ятого порядку зі змінною довжиною кроку за допомогою команди INTEGRATE.

Формат команди:

Name INTEGRATE A,[B],[C],[D],[E]

Name – ім'я змінної користувача. Операнд А – значення похідної змінної користувача за часом (права частина рівняння); операнд В – порогове значення першої змінної користувача; операнд С – мітка блока, до якого скеровується згенерований транзакт у момент, коли перша змінна користувача переходить через своє порогове значення в будь-якому напрямку; операнд D – порогове значення другої змінної користувача; операнд Е – мітка блока, до якого скеровується згенерований транзакт у момент, коли друга змінна користувача переходить через своє порогове значення в будь-якому напрямку.

Операнди А, В і D можуть бути іменем, числом, рядком, виразом у дужках або СЧА, а також викликами Plus-процедур.

5.1.1. Класична система з відмовами

Нагадаємо, що систему масового обслуговування називають *системою з відмовами*, якщо замовлення, яке надійшло в момент часу, коли всі канали зайняті, миттєво отримує відмову і покидає систему (втрачається).

Розглянемо n -канальну розімкнену СМО, на вхід якої подається найпростіший потік замовлень з інтенсивністю λ . Припустимо, що потік обслуговувань кожного каналу – також найпростіший з інтенсивністю μ . Якщо замовлення застає всі канали зайнятими, то покидає систему необслуженим. Якщо ж замовлення застає вільним хоча б один канал, то воно приймається на обслуговування будь-яким з вільних каналів і обслуговується до завершення.

Описану СМО називають *класичною*.

Побудуємо множину станів системи: x_0 – усі канали вільні, жодне замовлення не обслуговується; x_1 – зайнятий лише один канал, обслуговується одне замовлення; ...; x_k – зайнято k каналів, обслуговується k замовлень; ...; x_n – зайнято n каналів, обслуговується n замовлень. Граф станів цієї системи має вигляд:

$$x_0 \begin{array}{c} \lambda \\ \mu \end{array} \square x_1 \begin{array}{c} \lambda \\ 2\mu \end{array} \square \cdots \begin{array}{c} \lambda \\ k\mu \end{array} \square x_k \begin{array}{c} \lambda \\ (k+1)\mu \end{array} \square \cdots \begin{array}{c} \lambda \\ (n-1)\mu \end{array} \square x_{n-1} \begin{array}{c} \lambda \\ n\mu \end{array} \square x_n.$$

Ймовірність того, що система в момент часу t перебуває у стані x_i , позначимо через $p_i(t)$ ($i=0, 1 \dots n$). Використовуючи граф, запишемо систему диференціальних рівнянь для ймовірностей станів згідно з мнемонічним правилом [4]:

$$\begin{aligned} p'_0(t) &= -\lambda p_0(t) + \mu p_1(t); \\ &\dots\dots\dots \\ p'_k(t) &= -(\lambda + k\mu)p_k(t) + \lambda p_{k-1}(t) + (k+1)\mu p_{k+1}(t), \quad k = \overline{1, n-1}; \\ &\dots\dots\dots \\ p'_n(t) &= -n\mu p_n(t) + \lambda p_{n-1}(t). \end{aligned} \tag{5.1}$$

Розв'язок системи (5.1) з початковими умовами:

$$p_0(0) = 1; \quad p_k(0) = 0, \quad k = \overline{1, n}$$

повинен задовольняти нормувальну умову $\sum_{k=0}^n p_k(t) = 1, \quad t \geq 0$.

Граничний стаціонарний процес (стаціонарний режим роботи) ($t \rightarrow \infty$) для такої СМО існує, оскільки в ній немає станів без виходу і без входу, і всі потоки найпростіші. Для ймовірностей станів стаціонарного режиму отримаємо систему алгебричних рівнянь:

$$\begin{aligned} -\lambda p_0 + \mu p_1 &= 0; \\ &\dots\dots\dots \\ -(\lambda + k\mu)p_k + \lambda p_{k-1} + (k+1)\mu p_{k+1} &= 0, \quad k = \overline{1, n-1}; \\ &\dots\dots\dots \\ -n\mu p_n + \lambda p_{n-1} &= 0, \end{aligned} \tag{5.2}$$

яку розв'язують разом з нормувальною умовою $\sum_{k=0}^n p_k = 1$.

Ймовірності станів стаціонарного режиму (розв'язки системи (5.2)) визначаються за формулами Ерланга:

$$p_k = \frac{1}{k!} \left(\frac{\lambda}{\mu} \right)^k, \quad k = \overline{0, n}.$$

Ввівши позначення $\alpha = \lambda/\mu$ – середня кількість замовлень, які надходять у систему за середній час обслуговування одного замовлення в одному каналі, запишемо формули Ерланга у вигляді:

$$p_k = \frac{\frac{\alpha^k}{k!} e^{-\alpha}}{\sum_{k=0}^n \frac{\alpha^k}{k!} e^{-\alpha}} = \frac{P(k, \alpha)}{R(n, \alpha)}, \quad k = \overline{0, n}, \quad (5.3)$$

де $P(k, \alpha)$, $R(n, \alpha)$ – табличні функції пуассонівського розподілу (див. [4], § 2.10).

Зауважимо, що формули (5.3) залишаються правильними й тоді, коли час обслуговування має довільний розподіл з математичним сподіванням $\bar{t}_s = 1/\mu$.

Визначимо характеристики роботи класичної СМО з відмовами. Імовірність обслуговування замовлення $P_{обс}$, очевидно, дорівнює імовірності того, що замовлення, яке надійшло в систему, застане вільним хоча б один канал. Оскільки p_n – це ймовірність того, що зайняті всі n каналів, то:

$$P_{обс} = 1 - p_n = 1 - \frac{P(n, \alpha)}{R(n, \alpha)} = \frac{R(n, \alpha) - P(n, \alpha)}{R(n, \alpha)} = \frac{R(n-1, \alpha)}{R(n, \alpha)}. \quad (5.4)$$

З другого боку, ймовірність обслуговування замовлення дорівнює відносній пропускній здатності системи:

$$P_{обс} = \frac{\lambda_0}{\lambda} = \frac{\mu \bar{k}}{\lambda} = \frac{\bar{k}}{\alpha},$$

де λ_0 – щільність потоку обслужених замовлень (абсолютна пропускна здатність системи), \bar{k} – середня кількість зайнятих каналів. Отже,

$$\bar{k} = \alpha P_{обс} = \alpha \frac{R(n-1, \alpha)}{R(n, \alpha)}.$$

Вираз для \bar{k} можна отримати й безпосередньо через імовірності p_k (як математичне сподівання дискретної випадкової величини):

$$\bar{k} = \sum_{k=0}^n k p_k = \sum_{k=0}^n k \frac{P(k, \alpha)}{R(n, \alpha)} = \frac{1}{R(n, \alpha)} \sum_{k=0}^n k P(k, \alpha). \quad (5.5)$$

Імовірність того, що канал зайнятий, дорівнює відношенню середньої кількості зайнятих каналів \bar{k} до загальної кількості каналів n :

$$\pi_{з.к.} = \frac{\bar{k}}{n} = \frac{\alpha}{n} \frac{R(n-1, \alpha)}{R(n, \alpha)}. \quad (5.6)$$

Ця величина виводиться у стандартному звіті GPSS як UTIL. – коефіцієнт використання багатоканального пристрою.

Приклад 5.1. Побудувати модель чотириканальної СМО з відмовами, передбачивши можливість числового інтегрування системи диференціальних рівнянь для ймовірностей станів і порівняння результатів, отриманих за допомогою аналітичної та імітаційної моделей.

; Приклад 5.1.

```
Lam EQU 1/2 ; інтенсивність потоку замовлень
Mu EQU 1/10 ; інтенсивність потоку обслуговувань
lmo VARIABLE N$MIT2/N$MIT1 ; імовірність обслуговування
SYS STORAGE 4 ; 4-канальна СМО
CHMOD EQU 30 ; час моделювання
```

; Інтегрування системи диф. рівнянь для ймовірностей станів

```

p0_ EQU 1 ; початкові умови
p1_ EQU 0
p2_ EQU 0
p3_ EQU 0
p4_ EQU 0
p0_ INTEGRATE (-Lam#p0_+Mu#p1_)
p1_ INTEGRATE (-(Lam+Mu)#p1_+Lam#p0_+2#Mu#p2_)
p2_ INTEGRATE (-(Lam+2#Mu)#p2_+Lam#p1_+3#Mu#p3_)
p3_ INTEGRATE (-(Lam+3#Mu)#p3_+Lam#p2_+4#Mu#p4_)
p4_ INTEGRATE (-4#Mu#p4_+Lam#p3_)
*****
; Імітаційна модель СМО
GENERATE (Exponential(5,0,(1/Lam)))
MIT1 GATE SNF SYS,OUT ; перевірка умови незаповненості БКП
ENTER SYS
ADVANCE (Exponential(3,0,(1/Mu)))
MIT2 LEAVE SYS
OUT TERMINATE
GENERATE CHMOD
; Порівняння результатів *****
SAVEVALUE Imo,V$Imo ; імовірність обслуговування
; (імітаційна модель)
SAVEVALUE TImo,(1-p4_) ; мовірність обслуговування
; (аналітична модель)
; Середня кількість зайнятих каналів (аналітична модель)
SAVEVALUE SKZK,(p1_+2#p2_+3#p3_+4#p4_)
; Коефіцієнт використання БКП (аналітична модель)
SAVEVALUE KVYK,((p1_+2#p2_+3#p3_+4#p4_)/4)
TERMINATE 1

```

Ми розглядаємо СМО з відмовами з найпростішими потоками замовлень і обслуговувань з інтенсивностями $\lambda=1/2$, $\mu=1/10$ відповідно. Ці потоки моделюються за допомогою блоків GENERATE і ADVANCE відповідно та вмонтованих генераторів показникового розподілу. Наявність вільних каналів у момент надходження замовлення перевіряється блоком GATE. Якщо вільних каналів немає, то транзакт скеровується до блока TERMINATE з міткою OUT для знищення.

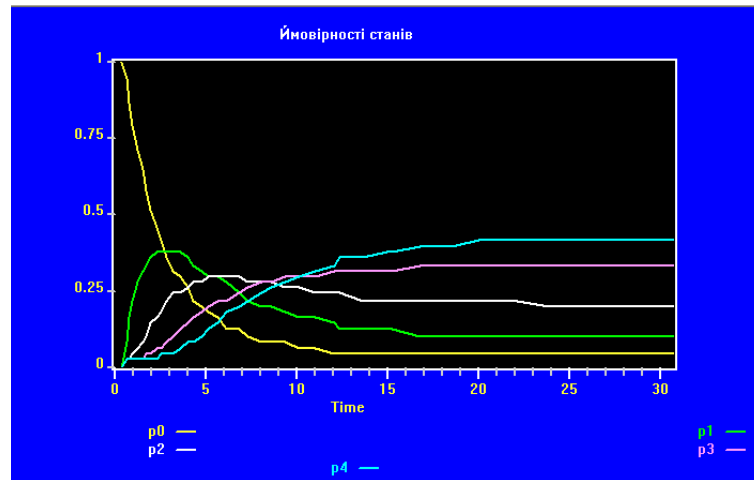


Рис. 5.1. Залежності ймовірностей станів від часу, отримані числовим інтегруванням рівнянь аналітичної моделі

Графічні залежності ймовірностей станів системи від часу, побудовані засобами GPSS World (у вікні Plot) за результатами числового інтегрування системи п'яти диференціальних рівнянь, отриманої з (5.1) при $n=4$, для значення часу моделювання 30, наведені на рис. 5.1.

Графік показує, що на проміжку модельного часу (20;25) згідно з результатами аналітичного моделювання має досягатися стаціонарний режим роботи системи.

Ймовірність обслуговування в імітаційній моделі обчислено як відношення кількості обслужених замовлень (транзактів, які пройшли через блок з міткою MIT2) до загальної кількості замовлень (транзактів, які пройшли через блок з міткою MIT1). В аналітичній моделі ймовірність обслуговування підраховуємо (див. формулу(5.4)) як різницю $1-p4_$, де $p4_$ ($p_4(t)$) – розв'язок системи диференціальних рівнянь для ймовірностей станів, отриманий шляхом числового інтегрування. Порівняння цих значень ймовірності обслуговування на проміжку модельного часу (0;200) зображено на рис. 5.2.

Рисунок показує, що на початковому етапі роботи моделі модельне значення ймовірності обслуговування (крива Pmod) дорівнює нулю, тому що перший транзакт проходить через блок з міткою MIT2 через деякий час після затримки у блоці ADVANCE.

Середню кількість зайнятих каналів в імітаційній моделі можна визначити за допомогою СЧА SASSYS, а в аналітичній моделі (див. формулу (5.5)) за допомогою виразу

$$p1_+2\#p2_+3\#p3_+4\#p4_$$

значення якого, як і раніше, можна обчислити, використовуючи числовий розв'язок системи диференціальних рівнянь для ймовірностей станів. На рис. 5.3 показано, як змінюються ці значення середньої кількості зайнятих каналів на проміжку модельного часу (0;200).

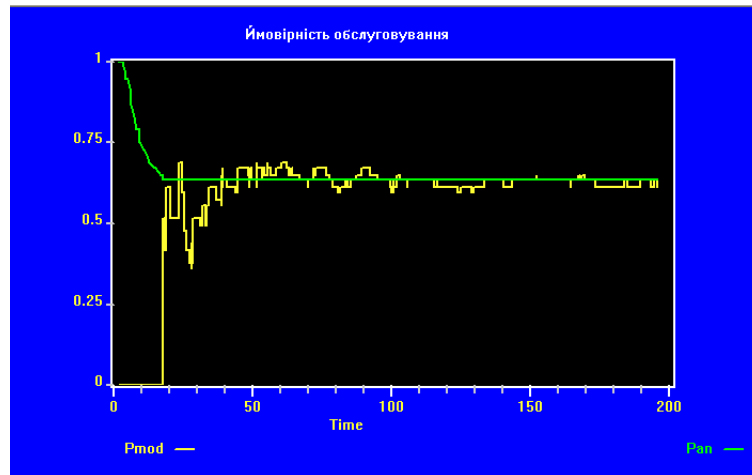


Рис. 5.2. Порівняння залежностей від часу модельного (Pmod) й аналітичного (Pan) значень ймовірності обслуговування

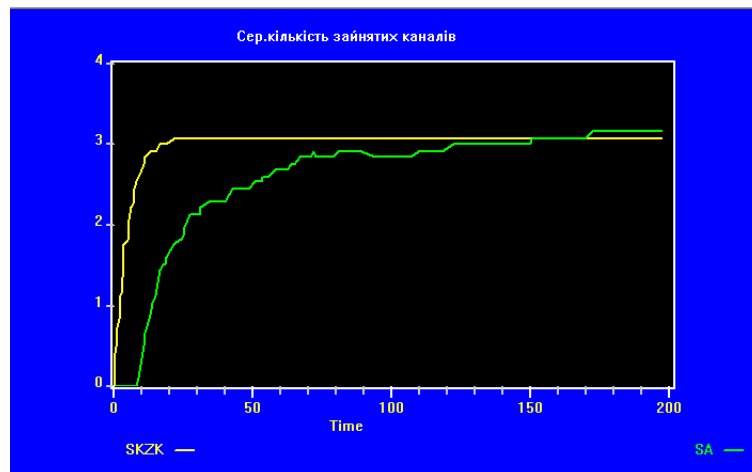


Рис. 5.3. Порівняння залежностей від часу модельного (SA) й аналітичного (SKZK) значень середньої кількості зайнятих каналів

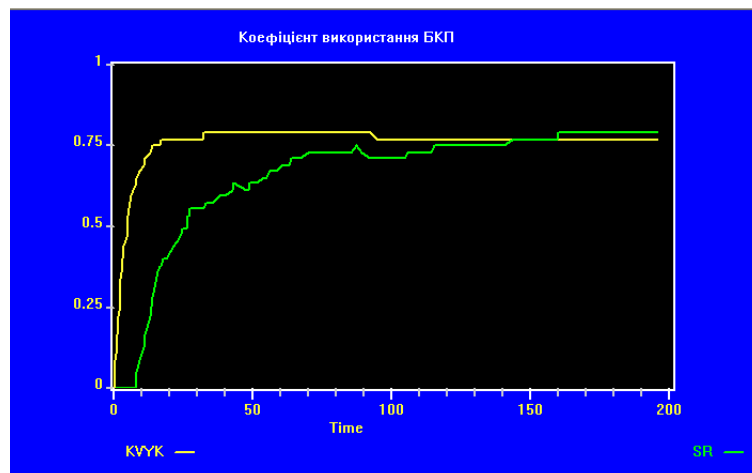


Рис. 5.4. Порівняння залежностей від часу модельного (SR) й аналітичного (KVYK) значень коефіцієнта використання БКП

Для визначення коефіцієнта використання СМО в імітаційній моделі слугує СЧА $SR\$SYS/1000$, а згідно з аналітичною моделлю потрібно скористатися формулою (5.6), тобто обчислене раніше значення середньої кількості зайнятих каналів поділити на чотири (кількість каналів СМО). Отримані графічні залежності показані на рис. 5.4.

На графіках (рис. 5.2-5.4) зі зростанням часу моделювання виразно простежується тенденція до зближення показників роботи СМО з відмовами, обчислених за результатами імітаційного та аналітичного моделювання. У моделі прикладу 5.1 передбачено також можливість отримання числових значень цих показників у стандартному звіті, тобто в момент завершення моделювання. Нижче наведені фрагменти стандартних звітів, отриманих для значень часу моделювання (CHMOD), рівних 200 і 100000.

NAME	VALUE
CHMOD	200.000
P0_	0.015
P1_	0.076
P2_	0.191
P3_	0.319
P4_	0.398

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SYS	4	0	0	4	62	1	3.115	0.779

SAVEVALUE	RETRY	VALUE
IMO	0	0.598
TIMO	0	0.602
SKZK	0	3.008
KVYK	0	0.752

NAME	VALUE
CHMOD	100000.000
P0_	0.015
P1_	0.076
P2_	0.191
P3_	0.319
P4_	0.398

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SYS	4	2	0	4	30005	1	3.012	0.753

SAVEVALUE	RETRY	VALUE
IMO	0	0.600
TIMO	0	0.602
SKZK	0	3.008
KVYK	0	0.752

Для обидвох значень часу моделювання розв'язки системи диференціальних рівнянь для ймовірностей станів $(p_{0_}, \dots, p_{4_})$ збігаються і дорівнюють стаціонарним значенням імовірностей станів (розв'язкам системи алгебричних рівнянь (5.2) ($n=4$), які визначені формулами (5.3)).

Інші показники роботи СМО з відмовами, обчислені за результатами імітаційного та аналітичного моделювання, для значення часу моделювання, який дорівнює 100000, дуже близькі. Зокрема, значення ймовірності обслуговування, збережені у відповідних комірках ІМО і ТІМО, дорівнюють відповідно 0,600 і 0,602; значення середньої кількості зайнятих каналів (AVE.C. і SKZK) становлять 3,012 і 3,008, а коефіцієнти використання БКП (UTIL. і KVYK) дорівнюють відповідно 0,753 і 0,752.

5.1.2. Система з обмеженням на довжину черги

Нехай до n -канальної СМО надходить найпростіший потік замовлень з інтенсивністю λ . Інтенсивність найпростішого потоку обслуговувань кожного каналу дорівнює μ . Якщо замовлення застає всі канали зайнятими, то воно стає в чергу і чекає свого обслуговування, не покидаючи чергу. Максимальна кількість місць у черзі – m . Кожне замовлення обслуговує лише один канал. Параметрами такої СМО з очікуванням є n, λ, μ і m .

Стани системи позначимо відповідно до кількості замовлень, які перебувають у системі (на обслуговуванні і в черзі): x_k ($k = \overline{0, n}$) – у системі є k замовлень, їх обслуговують k каналів, черги немає; x_{n+r} – у системі є $n+r$ замовлень ($r = \overline{1, m}$), n з них обслуговуються в n каналах і r замовлень перебуває в черзі. Отже, система має $n+m+1$ стан:

$$x_0 \overset{\lambda}{\square} \underset{\mu}{\square} x_1 \overset{\lambda}{\square} \underset{2\mu}{\square} \cdots \overset{\lambda}{\square} \underset{(n-1)\mu}{\square} x_{n-1} \overset{\lambda}{\square} \underset{n\mu}{\square} x_n \overset{\lambda}{\square} \underset{n\mu}{\square} x_{n+1} \overset{\lambda}{\square} \underset{n\mu}{\square} \cdots \overset{\lambda}{\square} \underset{n\mu}{\square} x_{n+m-1} \overset{\lambda}{\square} \underset{n\mu}{\square} x_{n+m}.$$

Від стану x_0 до стану x_n граф станів системи такий самий, як граф станів класичної СМО з відмовами. Для станів $x_{n+1}, x_{n+2}, \dots, x_{n+m-1}$ на систему діє потік звільнень зайнятих n каналів інтенсивності $n\mu$, намагаючись перевести її до сусіднього стану ліворуч, і потік замовлень інтенсивності λ .

Імовірність того, що система в момент часу t перебуває у стані x_i , позначимо через $p_i(t)$ ($i=0, 1 \dots n+m$). Використовуючи побудований граф, за допомогою мнемонічного правила можна записати систему диференціальних рівнянь для ймовірностей станів $p_i(t)$ ($i=0, 1 \dots n+m$).

Розв'язавши відповідну систему алгебричних рівнянь, отримаємо формули для ймовірностей станів стаціонарного режиму роботи системи ($t \rightarrow \infty$):

$$p_k = \frac{P(k, \alpha)}{R(n, \alpha) + \frac{\kappa(1-\kappa^m)}{1-\kappa} P(n, \alpha)} \quad (k = \overline{0, n});$$

$$p_{n+r} = \kappa^r p_n \quad (r = \overline{1, m}); \quad \alpha = \frac{\lambda}{\mu}; \quad \kappa = \frac{\alpha}{n} \neq 1.$$

Імовірність обслуговування замовлення дорівнює імовірності того, що замовлення, яке надійшло в систему, застане вільним хоча б один з каналів або хоча б одне місце в черзі:

$$P_{\text{обс}} = 1 - p_{n+m} = 1 - \kappa^m p_n. \quad (5.7)$$

Середню кількість зайнятих каналів і середню кількість замовлень, які перебувають у черзі, обчислюють як математичні сподівання:

$$\bar{k} = \sum_{k=0}^n k p_k + n \sum_{r=1}^m p_{n+r}; \quad (5.8)$$

$$\bar{r} = \sum_{r=1}^m r p_{n+r} = p_n \sum_{r=1}^m r \kappa^r. \quad (5.9)$$

Приклад 5.2. Побудувати модель чотириканальної СМО з двома місцями у черзі, передбачивши можливість числового інтегрування системи диференціальних рівнянь для ймовірностей станів, порівняння результатів, отриманих за допомогою аналітичної та імітаційної моделей, і табулювання часу перебування замовлень у черзі й у системі загалом.

; Приклад 5.2.

```
Lam EQU 1/2 ; інтенсивність потоку замовлень
Mu EQU 1/10 ; інтенсивність потоку обслуговувань
Em EQU 2 ; кількість місць у черзі
Imo VARIABLE N$MIT2/N$MIT1 ; імовірність обслуговування
WT QTABLE CHER,0,2,11 ; таблиця розподілу часу в черзі
ALLT TABLE M1,0,5,13 ; таблиця розподілу часу в системі
SYS STORAGE 4 ; 4-канальна СМО
```

; Інтегрування системи диф. рівнянь для ймовірностей станів

```
p0_ EQU 1 ; початкові умови
```

```
p1_ EQU 0
```

```
p2_ EQU 0
```

```
p3_ EQU 0
```

```
p4_ EQU 0
```

```
p5_ EQU 0
```

```
p6_ EQU 0
```

```
p0_ INTEGRATE (-Lam#p0_+Mu#p1_)
```

```
p1_ INTEGRATE (-(Lam+Mu)#p1_+Lam#p0_+2#Mu#p2_)
```

```
p2_ INTEGRATE (-(Lam+2#Mu)#p2_+Lam#p1_+3#Mu#p3_)
```

```
p3_ INTEGRATE (-(Lam+3#Mu)#p3_+Lam#p2_+4#Mu#p4_)
```

```
p4_ INTEGRATE (-(Lam+4#Mu)#p4_+Lam#p3_+4#Mu#p5_)
```

```
p5_ INTEGRATE (-(Lam+4#Mu)#p5_+Lam#p4_+4#Mu#p6_)
```

```
p6_ INTEGRATE (Lam#p5_-4#Mu#p6_)
```

; Імітаційна модель СМО

```
GENERATE (Exponential(1,0,(1/Lam)))
```

```

MIT1    TEST L Q$CHER,Em,OUT ; чи є вільне місце в черзі
        QUEUE CHER
        ENTER SYS
        DEPART CHER
        ADVANCE (Exponential(1,0,(1/Mu)))
MIT2    LEAVE SYS
OUT     TABULATE ALLT ; табулювати час перебування в системі
        TERMINATE
        GENERATE 30
; Порівняння результатів *****
SAVEVALUE lmo,V$lmo
SAVEVALUE Tlmo,(1-p6_)
SAVEVALUE SKZK,(p1_+2#p2_+3#p3_+4#(p4_+p5_+p6_))
SAVEVALUE KVKYK,((p1_+2#p2_+3#p3_+4#(p4_+p5_+p6_))/4)
SAVEVALUE SKZCH,(p5_+2#p6_)
        TERMINATE 1

```

Ми розглядаємо СМО з двома місцями у черзі з найпростішими потоками замовлень і обслуговувань з тими самими інтенсивностями $\lambda=1/2$, $\mu=1/10$, що й у прикладі 5.1. Наявність вільних місць у черзі в момент надходження замовлення перевіряє блок TEST. Якщо вільних місць немає, то транзакт скеровується до блока з міткою OUT і після внесення інформації у статистику часу перебування в системі (блок TABULATE) передається до блока TERMINATE для знищення.

У моделі здійснюється числове інтегрування системи семи диференціальних рівнянь для ймовірностей станів, яка записана за допомогою графа станів ($n=4$, $m=2$). Графічні залежності ймовірностей станів від часу, отримані у вікні Plot за результатами числового інтегрування, наведені на рис. 5.5.

Графік показує, що на проміжку модельного часу (20;25), як і за результатами аналітичного моделювання для прикладу 5.1, має досягатися стаціонарний режим роботи системи.

Ймовірність обслуговування в імітаційній моделі, як і раніше, обчислена як відношення кількості обслужених замовлень до загальної кількості замовлень. В аналітичній моделі ймовірність обслуговування підраховуємо (див. формулу (5.7)) як різницю $1-p6_$, де $p6_$ – числовий розв’язок системи диференціальних рівнянь для ймовірностей станів. Зближення цих значень ймовірності обслуговування (див. рис. 5.6) починається тоді, коли модельний час перевищує значення 50.

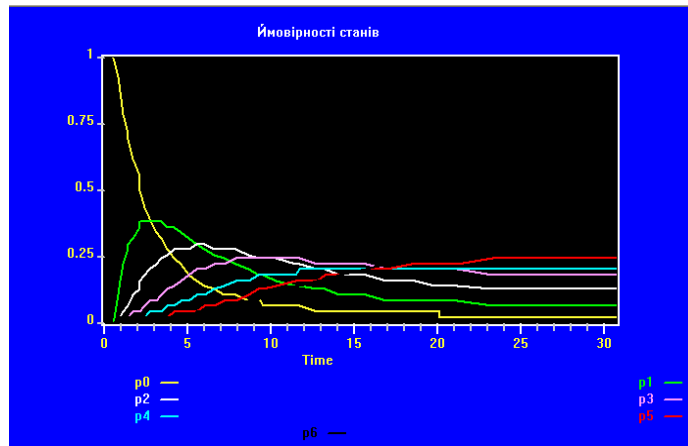


Рис. 5.5. Залежності ймовірностей станів від часу, отримані числовим інтегруванням рівнянь аналітичної моделі

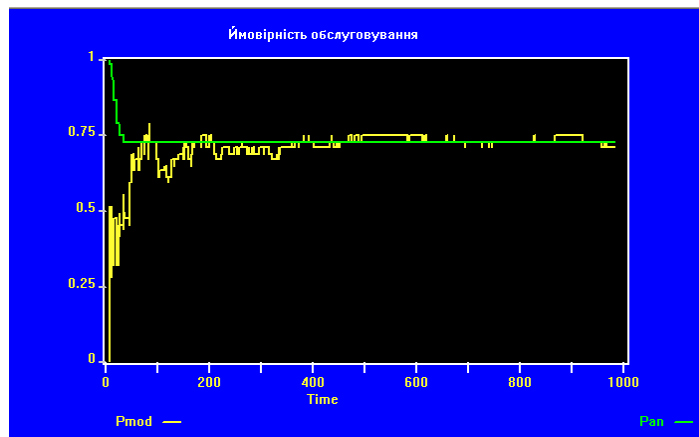


Рис. 5.6. Порівняння залежностей від часу модельного (Pmod) й аналітичного (Pan) значень ймовірності обслуговування

Середню кількість замовлень, які перебувають у черзі, в імітаційній моделі можна визначити за допомогою СЧА QASHER, а в аналітичній моделі (див. формулу (5.9)) за допомогою виразу $p5_{+2} \# p6_{+}$, значення якого, як і раніше, можна обчислити, використовуючи числовий розв'язок системи диференціальних рівнянь для ймовірностей станів. На рис. 5.7 показано, як змінюються ці значення середньої кількості замовлень, що перебувають у черзі, на проміжку модельного часу (0;2000).

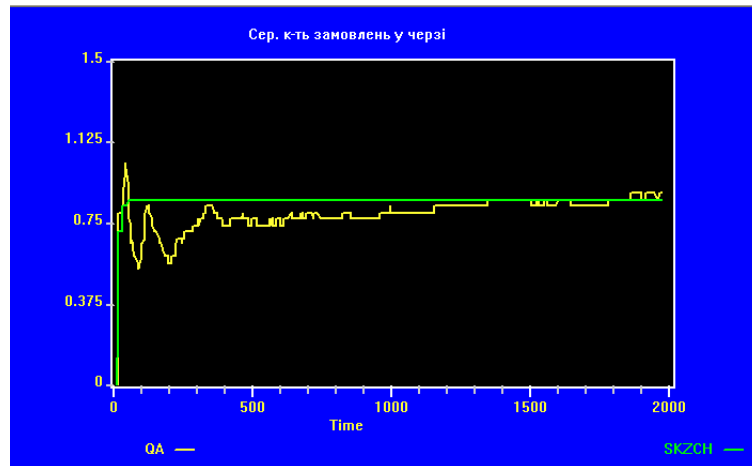


Рис. 5.7. Порівняння залежностей від часу модельного (QA) й аналітичного (SKZCH) значень середньої кількості замовлень у черзі

Оператори QTABLE і TABLE з іменами WT і ALLT відповідно визначають таблиці розподілу часу перебування в черзі і в системі загалом. Верхня межа першого частотного інтервалу таблиць дорівнює нулеві, що дає змогу окремо виділити замовлення з „нульовим” часом очікування (в першому інтервалі таблиці WT) і замовлення, які, не потрапивши на обслуговування, покинули систему (в першому інтервалі таблиці ALLT). Самі таблиці будуть виведені в стандартному звіті моделі, а їхню візуальну інтерпретацію (гістограми відповідних розподілів) можна побачити у вікні TABLE (див. рис. 5.8-5.9, які відповідають значенню часу моделювання, який дорівнює 10000). У верхній частині вікна зазначено середнє значення і середнє квадратичне відхилення табульованого аргументу.

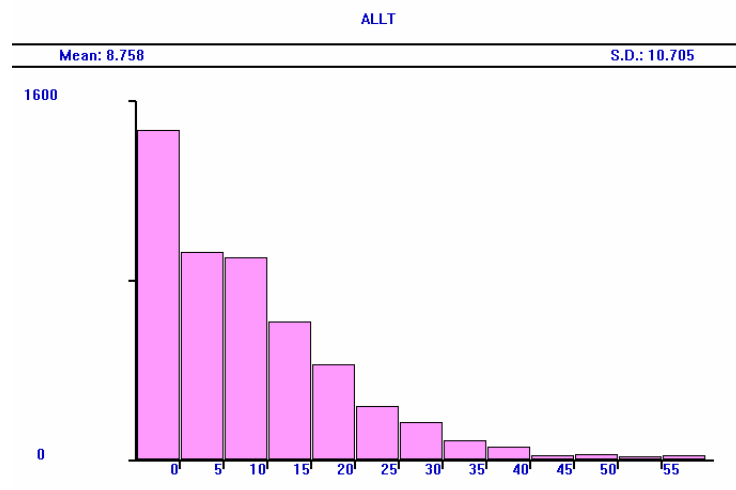


Рис. 5.8. Гістограма розподілу часу перебування замовлення в системі з урахуванням втрачених замовлень

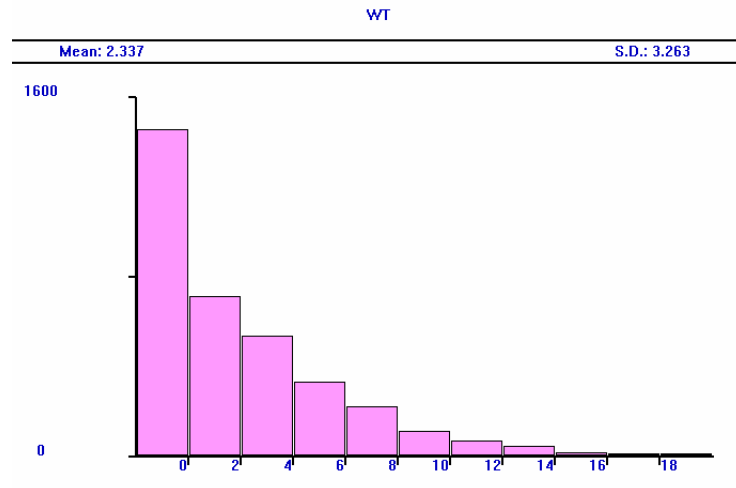


Рис. 5.9. Гістограма розподілу часу перебування замовлення в черзі

Для того щоб отримати гістограму розподілу часу перебування замовлення в системі без урахування втрачених замовлень потрібно змінити розташування мітки OUT, перенісши її від блока TABULATE до блока TERMINATE. Тоді транзакти, які не потрапляють до блока ENTER, будуть скеровані до блока TERMINATE, оминувши блок TABULATE. У цьому випадку (див. рис. 5.10) бажано встановити верхню межу першого частотного інтервалу таблиці ALLT такою, яка дорівнює, наприклад, 5, інакше перший інтервал гістограми буде порожнім.

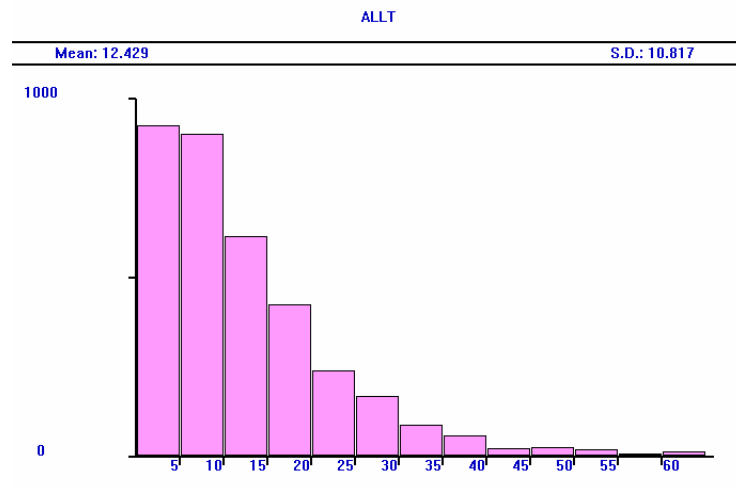


Рис. 5.10. Гістограма розподілу часу перебування замовлення в системі без урахування втрачених замовлень

Інші показники роботи СМО з очікуванням, обчислені за результатами імітаційного та аналітичного моделювання, і виведені у стандартному звіті моделі для значення часу моделювання, яке дорівнює 10000, дуже близькі. Зокрема, значення ймовірності обслуговування, збережені у відповідних комірках ІМО і ТІМО, дорівнюють відповідно 0,704 і 0,706; значення середньої кількості зайнятих каналів (AVE.C. і SKZK) становлять 3,543 і 3,532 (останнє з них обчислене за формулою (5.8)), коефіцієнти використання БКП (UTIL. і

KVYK) дорівнюють відповідно 0,886 і 0,883, а значення середньої кількості замовлень у черзі (AVE.CONT. і SKZCH) становлять 0,820 і 0,822. Середній час очікування в черзі (AVE.TIME) дорівнює 2,337 одиниць модельного часу і збігається із середнім значенням табульованого аргументу гістограми WT (рис. 5.9).

5.1.3. Система без обмежень на довжину черги

Ми вивчали СМО з очікуванням у припущенні, що кількість місць у черзі – обмежена ($m < \infty$). Тепер розглянемо систему без обмежень на довжину черги ($m = \infty$). Для такої системи стаціонарний режим існує лише за умови, що $\kappa = \frac{\lambda}{n\mu} = \frac{\alpha}{n} < 1$.

Для одноканальної системи без обмежень на довжину черги стаціонарний режим існує за умови, що $\alpha < 1$. Показники ефективності системи у стаціонарному режимі визначають так [4]:

$$\bar{k} = \alpha, \quad P_{\text{обс}} = 1, \quad \bar{r} = \frac{\alpha^2}{1-\alpha}, \quad (5.10)$$

де \bar{k} – середня кількість зайнятих каналів, $P_{\text{обс}}$ – ймовірність обслуговування; \bar{r} – середня довжина черги.

Якщо W – час очікування в черзі, то згідно з [4] математичні сподівання цієї випадкової величини у стаціонарному режимі для одноканальної СМО обчислюють за формулами:

$$M(W|W > 0) = \frac{1}{(1-\alpha)\mu}; \quad M(W) = \frac{\alpha}{(1-\alpha)\mu}, \quad (5.11)$$

де $M(W|W > 0)$ – середній час очікування без урахування нульових входжень.

Приклад 5.3. Побудувати модель одноканальної СМО з необмеженою довжиною черги, передбачивши можливість порівняння результатів, отриманих за допомогою аналітичної моделі для стаціонарного режиму та імітаційної моделі.

; Приклад 5.3.

```
LamEQU 1/3           ; інтенсивність потоку замовлень
Mu EQU 1/2           ; інтенсивність потоку обслуговувань
ALF EQU Lam/Mu       ; середня кількість зайнятих каналів (див. (5.10))
WA EQU 1/((1-ALF)#Mu) ; середній час очікування без нульових
                       ; входжень (див. (5.11))
WA0 EQU WA#ALF       ; середній час очікування (див. (5.11))
RR EQU (ALF^2)/(1-ALF) ; середня довжина черги (див. (5.10))
SYSSTORAGE 1         ; одноканальна СМО
*****
GENERATE (Exponential(1,0,(1/Lam)))
```

```

QUEUE CHER
ENTER SYS
DEPART CHER
ADVANCE (Exponential(1,0,(1/Mu)))
LEAVE SYS
TERMINATE
GENERATE 5000      ; час моделювання
TERMINATE 1
    
```

Ми розглядаємо СМО з найпростішими потоками замовлень і обслуговувань з інтенсивностями $\lambda=1/3$, $\mu=1/2$ відповідно. Змінні користувача Lam, Mu, ALF, WA, WA0 і RR введені в моделі з метою обчислення стаціонарних значень середньої кількості зайнятих каналів, середнього часу очікування в черзі без урахування і з урахуванням нульових входжень у чергу та середньої довжини черги, які відповідають аналітичній моделі даної СМО. На графіках (рис. 5.11-5.14) ці значення зображені прямими, паралельними осі абсцис (осі часу), і позначають еталонні стаціонарні значення, до яких мають із плином модельного часу наблизитися відповідні показники, отримані за допомогою імітаційної моделі.

Для одержання останніх потрібно ввести в полі Expression вікна Edit Plot відповідні СЧА: S\$SYS – середня кількість зайнятих каналів; Q\$CHER – середня довжина черги; QX\$CHER – середній час очікування без урахування нульових входжень; QT\$CHER – середній час очікування з урахуванням нульових входжень.

Рисунки показують, що стабілізація цих показників і наближення їхніх значень до відповідних стаціонарних настає після тривалого часу роботи системи. Динаміку зближення можна простежити за даними таблиці 5.1.



Рис. 5.11. Порівняння модельного (SA) і стаціонарного аналітичного (SKZK) значень середньої кількості зайнятих каналів

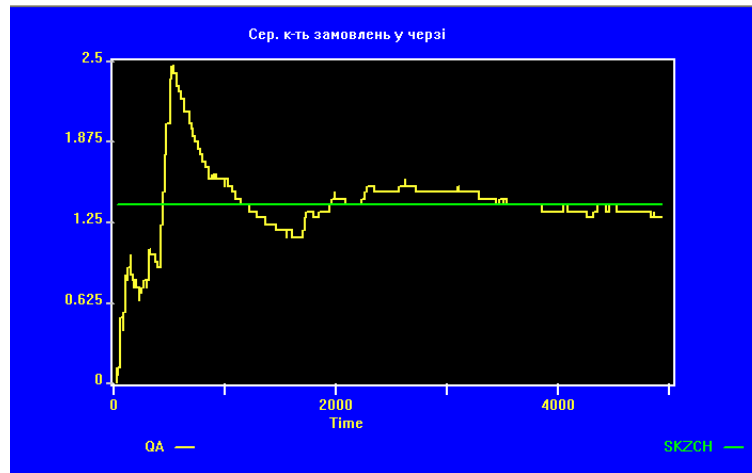


Рис. 5.12. Порівняння модельного (QA) і стаціонарного аналітичного (SKZCH) значень середньої довжини черги

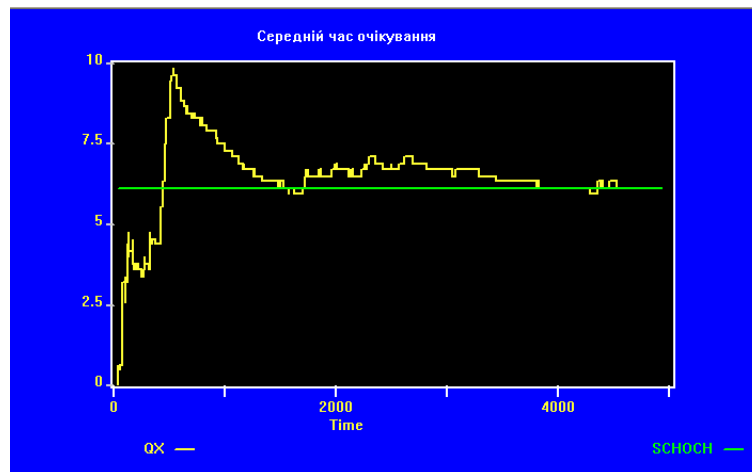


Рис. 5.13. Порівняння модельного (QX) і стаціонарного аналітичного (SCHOCH) значень середнього часу очікування без урахування нульових входжень

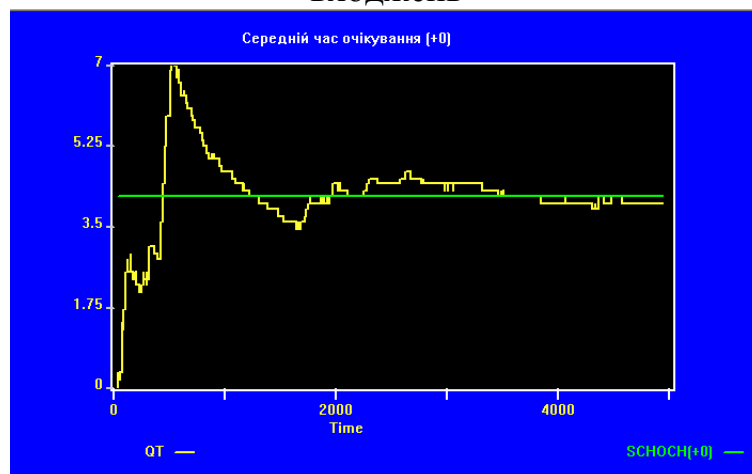


Рис. 5.14. Порівняння модельного (QT) і стаціонарного аналітичного (SCHOCH(+0)) значень середнього часу очікування з урахуванням нульових входжень

Таблиця 5.1

Час моделювання	Середня кількість зайнятих каналів (імітаційна модель)	Середня довжина черги (імітаційна модель)	Середній час очікування (- 0) (імітаційна модель)	Середній час очікування (імітаційна модель)
100	0,693	0,630	3,316	1,800
1000	0,657	1,512	7,202	4,611
5000	0,674	1,258	5,865	3,839
10000	0,691	1,890	8,170	5,624
100000	0,672	1,381	6,175	4,146
1000000	0,668	1,337	5,993	4,010
2000000	0,667	1,331	5,990	4,001
Стаціонарне значення (аналіт. модель)	0,667	1,333	6,000	4,000

5.2. Система масового обслуговування G/G/1/0

5.2.1. Аналітична модель

Розглянемо одноканальну СМО з відмовами. Припустимо, що потік замовлень і потік обслуговувань є стаціонарними ординарними потоками, а випадкові величини T_λ (час між моментами надходження замовлень) і T_μ (час обслуговування одного замовлення) мають скінченні математичні сподівання: $m_\lambda = M(T_\lambda)$, $m_\mu = M(T_\mu)$.

Проаналізуємо роботу системи на дуже великому проміжку часу T . За початок відліку часу приймемо момент надходження першого замовлення і зупинимо відлік часу в момент завершення обслуговування чергового замовлення.

Процес функціонування системи є чергуванням випадкових проміжків часу тривалістю T_μ і T_0 (час від моменту завершення обслуговування чергового замовлення до моменту прибуття наступного). Якщо N – кількість замовлень, які надійшли до системи за час T , $N_{обс}$ – кількість обслужених замовлень за цей час, то виконується наближена рівність

$$T \approx (N-1)m_\lambda + \tilde{m}_\mu \approx N_{обс}(m_\mu + m_0) - m_0, \quad (5.12)$$

де $m_0 = M(T_0)$; \tilde{m}_μ – час від моменту надходження останнього замовлення до завершення часу T . Якщо останнє замовлення, яке надійшло за час T , встигло пройти обслуговування, то в (5.12) потрібно прийняти $\tilde{m}_\mu = m_\mu$.

Рівність (5.12) виконується тим точніше, чим триваліший проміжок часу T розглядається (чим більше N). Виразивши з (5.12) відношення $N_{обс}/N$ і перейшовши до границі при $N \rightarrow \infty$, визначимо ймовірність обслуговування замовлення, яке надійшло до системи,

$$P_{обс} = \lim_{N \rightarrow \infty} \frac{N_{обс}}{N} = \lim_{N \rightarrow \infty} \frac{m_\lambda + \frac{\tilde{m}_\mu + m_0 - m_\lambda}{N}}{m_\mu + m_0} = \frac{m_\lambda}{m_\mu + m_0}. \quad (5.13)$$

Отриманий результат не зміниться, якщо за початок чи кінець проміжка часу T взяти будь-який інший момент функціонування системи, тому отримана формула для ймовірності обслуговування відповідає граничному стаціонарному процесу для одноканальної СМО з відмовами. Вигляд правої частини (5.13) дає змогу стверджувати, що умовою існування граничного стаціонарного процесу є існування скінченних математичних сподівань випадкових величин T_λ , T_μ і T_0 .

Розглядаючи рівність (5.12), зазначимо, що проміжок часу T складається із суми довжин проміжків T_μ (часу зайнятості системи) і суми довжин проміжків T_0 (часу, коли система простоє), тобто $T \approx \sum T_\mu + \sum T_0$, де

$$\sum T_\mu \approx N_{обс} m_\mu; \quad \sum T_0 \approx (N_{обс} - 1) m_0.$$

Пронумерувавши стани системи відповідно до кількості замовлень у системі

(x_0 – система вільна, x_1 – в системі є одне замовлення), для ймовірностей станів граничного стаціонарного процесу отримаємо формули:

$$p_0 = \lim_{N \rightarrow \infty} \frac{\sum T_0}{T} = \lim_{N \rightarrow \infty} \frac{\frac{N_{обс}}{N} m_0 - \frac{m_0}{N}}{m_\lambda + \frac{\tilde{m}_\mu}{N}} = \frac{m_0 P_{обс}}{m_\lambda} = \frac{m_0}{m_\mu + m_0};$$

$$p_1 = \lim_{N \rightarrow \infty} \frac{\sum T_\mu}{T} = \lim_{N \rightarrow \infty} \frac{\frac{N_{обс}}{N} m_\mu}{m_\lambda + \frac{\tilde{m}_\mu}{N}} = \frac{m_\mu P_{обс}}{m_\lambda} = \frac{m_\mu}{m_\mu + m_0},$$
(5.14)

де p_i – стаціонарна ймовірність перебування системи у стані x_i ($i=0, 1$).

Якщо потік замовлень найпростіший (стаціонарний ординарний потік з інтервалами часу між сусідніми подіями, розподіленими за показниковим законом з параметром λ), то внаслідок відсутності післядії випадкова величина T_0 також розподілена за показниковим законом з параметром λ . Отже, у цьому випадку $m_0 = m_\lambda = 1/\lambda$ і зі співвідношень (5.13), (5.14) отримаємо відомі формули для одноканальної системи з найпростішим потоком замовлень:

$$P_{обс} = p_0 = \frac{1}{\lambda m_\mu + 1}; \quad p_1 = \frac{\lambda m_\mu}{\lambda m_\mu + 1}.$$
(5.15)

Порівнюючи співвідношення (5.13), (5.14) і (5.15), можемо зробити висновок, що *лише для системи з найпростішим потоком замовлень $P_{обс} = p_0$.*

Проаналізуємо випадкову величину T_0 . Оскільки проміжки часу T_λ і T_μ починаються одночасно, то тривалість часу T_0 залежить від кількості замовлень, які надійшли до системи за час обслуговування одного замовлення T_μ . Маємо: $T_0 = T_\lambda - T_\mu$, якщо за час T_μ не надійшло жодного замовлення (подія A_1); $T_0 = T_\lambda + T_\lambda - T_\mu$, якщо за час T_μ надійшло одне замовлення (подія A_2); $T_0 = T_\lambda + T_\lambda + T_\lambda - T_\mu$, якщо за час T_μ надійшло два замовлення (подія A_3) і т. п. Імовірності подій A_k ($k=1,2,\dots$) відповідно рівні:

$$P(A_1) = q_1 = P\{T_\mu < T_\lambda\}; \quad P(A_2) = q_2 = P\{T_\lambda \leq T_\mu < T_\lambda + T_\lambda\};$$

$$P(A_3) = q_3 = P\{T_\lambda + T_\lambda \leq T_\mu < T_\lambda + T_\lambda + T_\lambda\}, \dots$$

Визначимо математичне сподівання випадкової величини T_0 : $M(T_0) = m_\lambda - m_\mu$ з імовірністю q_1 ; $M(T_0) = 2m_\lambda - m_\mu$ з імовірністю q_2 ; $M(T_0) = 3m_\lambda - m_\mu$ з імовірністю q_3 і т. п. Математичне сподівання m_0 можемо обчислити за формулою повного математичного сподівання:

$$m_0 = \sum_{k=1}^{\infty} (km_\lambda - m_\mu)q_k = m_\lambda \sum_{k=1}^{\infty} kq_k - m_\mu.$$

Тут враховано, що $\sum_{k=1}^{\infty} q_k = 1$, оскільки події A_k ($k=1,2,\dots$) утворюють повну групу попарно несумісних подій.

Підсумуємо викладені міркування.

Теорема 5.1. Якщо потік замовлень і потік обслуговувань є стаціонарними ординарними потоками зі скінченними математичними сподіваннями інтервалів часу між сусідніми подіями $M(T_\lambda) = m_\lambda$, $M(T_\mu) = m_\mu$, і збігається числовий ряд

$$S_q = \sum_{k=1}^{\infty} kq_k, \quad (5.16)$$

$$q_k = P\{\underbrace{T_\lambda + \dots + T_\lambda}_{(k-1)\text{ доданок}} \leq T_\mu < \underbrace{T_\lambda + \dots + T_\lambda}_k, \quad k=1,2,\dots,$$

то для одноканальної СМО з відмовами існує граничний стаціонарний процес, для якого ймовірність обслуговування і ймовірності станів системи визначають за формулами:

$$P_{обс} = \frac{m_\lambda}{m_\mu + m_0} = \frac{1}{S_q}, \quad p_0 = \frac{m_0}{m_\mu + m_0} = \frac{m_0}{m_\lambda} P_{обс},$$

$$p_1 = \frac{m_\mu}{m_\mu + m_0} = \frac{m_\mu}{m_\lambda} P_{обс}, \quad m_0 = m_\lambda S_q - m_\mu. \quad (5.17)$$

Оскільки $S_q = M(X) + 1$, де X – кількість замовлень, які надійшли до системи за час T_μ , то збіжність ряду (5.16) означає, що середня кількість замовлень, які надходять до системи за час обслуговування одного замовлення T_μ , є скінченною.

Наведемо результати обчислення суми ряду (5.16) і ймовірностей (5.17) для деяких типових розподілів випадкових величин T_λ і T_μ .

• Випадкові величини T_λ і T_μ розподілені за законами Ерланга другого порядку з параметрами λ і μ відповідно.

$$S_q = \mu^2 \sum_{k=1}^{\infty} \frac{k(2k\lambda + (2k-1)(\lambda + \mu))\lambda^{2k-2}}{(\lambda + \mu)^{2k+1}} =$$

$$= \frac{(\lambda + \mu)(4\lambda^2 + 3\lambda\mu + \mu^2)}{\mu(2\lambda + \mu)^2}; \quad P_{обс} = \frac{\mu(2\lambda + \mu)^2}{(\lambda + \mu)(4\lambda^2 + 3\lambda\mu + \mu^2)}, \quad (5.18)$$

$$\bar{k} = p_1 = \frac{\lambda(2\lambda + \mu)^2}{(\lambda + \mu)(4\lambda^2 + 3\lambda\mu + \mu^2)}, \quad p_0 = 1 - p_1,$$

де \bar{k} – середня кількість зайнятих каналів.

• Потік замовлень регулярний ($T_\lambda = T = \text{const}$), час обслуговування T_μ рівномірно розподілений на проміжку $(0, a)$.

Для $a \leq T$

$$q_1 = 1, \quad q_k = 0 \quad (k = 2, 3, \dots); \quad S_q = 1.$$

Для $(s-1)T \leq a \leq sT$ ($s=2, 3, \dots$)

$$q_k = \frac{T}{a}, \quad k = \overline{1, s-1}; \quad q_s = 1 - \frac{(s-1)T}{a};$$

$$q_k = 0, \quad k = s+1, s+2, \dots; \quad S_q = s - \frac{s(s-1)T}{2a};$$

$$P_{обс} = \frac{2a}{s(2a - (s-1)T)}, \quad \bar{k} = p_1 = \frac{a^2}{sT(2a - (s-1)T)}, \quad p_0 = 1 - p_1. \quad (5.19)$$

- Час T_λ розподілений за законом Ерланга другого порядку з параметром λ , потік обслуговувань регулярний ($T_\mu = T = \text{const}$).

$$S_q = e^{-\lambda T} \sum_{k=1}^{\infty} k \left(\frac{(\lambda T)^{2k-2}}{(2k-2)!} + \frac{(\lambda T)^{2k-1}}{(2k-1)!} \right) = \frac{1}{4}(3 + 2\lambda T + e^{-2\lambda T});$$

$$P_{\text{обс}} = \frac{4}{3 + 2\lambda T + e^{-2\lambda T}}, \quad \bar{k} = p_1 = \frac{2\lambda T}{3 + 2\lambda T + e^{-2\lambda T}}, \quad p_0 = 1 - p_1. \quad (5.20)$$

- Потік замовлень регулярний ($T_\lambda = T = \text{const}$), потік обслуговувань найпростіший ($T_\mu = 1/\mu$).

$$S_q = \sum_{k=1}^{\infty} k \left(e^{-(k-1)\mu T} - e^{-k\mu T} \right) e^{-\mu T} = \frac{1}{1 - e^{-\mu T}};$$

$$P_{\text{обс}} = 1 - e^{-\mu T}, \quad \bar{k} = p_1 = \frac{1}{\mu T} (1 - e^{-\mu T}), \quad p_0 = 1 - p_1. \quad (5.21)$$

5.2.2. Апробація аналітичної моделі за допомогою імітаційного моделювання

Приклад 5.4. Перевірити результати, отримані в п. 5.2.1 для чотирьох випадків задання потоків замовлень і обслуговувань, за допомогою імітаційної моделі одноканальної СМО з відмовами. Задати для різних розподілів однакові значення середнього інтервалу надходження замовлень – 4 од. модельного часу і середнього часу обслуговування – 5 од. модельного часу.

; Приклад 5.4.

La EQU 1/2 ; задання λ

Mu EQU 2/5 ; задання μ

Mu4EQU 1/5 ; задання μ_4

; аналітичні значення показників системи 1

P1CHYS EQU Mu#(2#La+Mu)^2

P1ZN EQU (La+Mu)#(4#La^2+3#La#Mu+Mu^2)

POBS1 EQU P1CHYS/P1ZN ; імовірність обслуговування системи 1

SKZK1 EQU (La/Mu)#POBS1 ; середня кількість зайнятих каналів ; системи 1

; аналітичні значення показників системи 2

a_ EQU 10

T2 EQU 4

s_ EQU 3

P2ZN EQU s_#(2#a_-(s_-1)#T2)

POBS2 EQU 2#a_/P2ZN ; імовірність обслуговування системи 2

SKZK2 EQU (a_/(2#T2))#POBS2 ; середня кількість зайнятих каналів ; системи 2

; аналітичні значення показників системи 3

```

T3      EQU 5
POBS3   EQU 4/(3+2#La#T3+EXP(-2#La#T3)) ; імовірність
                                               ; обслуговування системи 3
SKZK3   EQU (La#T3/2)#POBS3 ; середня кількість зайнятих каналів
                                               ; системи 3
; аналітичні значення показників системи 4
T4      EQU 4
Mu4     EQU 1/5
POBS4   EQU 1-EXP(-Mu4#T4) ; імовірність обслуговування системи 4
SKZK4   EQU POBS4/( Mu4#T4) ; середня кількість зайнятих каналів
                                               ; системи 4
; модельні значення ймовірностей обслуговування систем 1-4
Imo1    VARIABLE N$MIT21/N$MIT11
Imo2    VARIABLE N$MIT22/N$MIT12
Imo3    VARIABLE N$MIT23/N$MIT13
Imo4    VARIABLE N$MIT24/N$MIT14
SYS1    STORAGE 1                ; одноканальна СМО
SYS2    STORAGE 1                ; одноканальна СМО
SYS3    STORAGE 1                ; одноканальна СМО
SYS4    STORAGE 1                ; одноканальна СМО
CHMOD   EQU 5000                 ; час моделювання
*****
; 1) Випадкові величини  $T_\lambda$  і  $T_\mu$  розподілені за законами Ерланга
; другого порядку з параметрами  $\lambda=0,5$  і  $\mu=0,4$  відповідно
GENERATE (Exponential(5,0,(1/La))+Exponential(5,0,(1/La)))
MIT11   GATE SNF SYS1,OUT1 ; перевірка умови
                                               ; незаповненості ОКП
        ENTER SYS1
ADVANCE (Exponential(3,0,(1/Mu))+Exponential(3,0,(1/Mu)))
MIT21   LEAVE SYS1
OUT1    TERMINATE
*****
; 2) Потік замовлень регулярний ( $T_\lambda = T2=4$ ), час обслуговування
;  $T_\mu$  рівномірно розподілений на проміжку (0,10)
GENERATE T2
MIT12   GATE SNF SYS2,OUT2 ; перевірка умови
                                               ; незаповненості ОКП
        ENTER SYS2
ADVANCE (Uniform(3,0,10))
MIT22   LEAVE SYS2
OUT2    TERMINATE
*****

```

; 3) Час T_λ розподілений за законом Ерланга другого порядку з
; параметром $\lambda=0,5$, потік обслуговувань регулярний ($T_\mu = T3=5$)

GENERATE (Exponential(5,0,(1/La))+Exponential(5,0,(1/La)))

MIT13 GATE SNF SYS3,OUT3 ; перевірка умови

; незаповненості ОКП

ENTER SYS3

ADVANCE T3

MIT23 LEAVE SYS3

OUT3 TERMINATE

; 4) Потік замовлень регулярний ($T_\lambda = T4=4$), потік обслуговувань

; найпростіший з параметром $\mu_4 = 1/5$

GENERATE T4

MIT14 GATE SNF SYS4,OUT4 ; перевірка умови

; незаповненості ОКП

ENTER SYS4

ADVANCE (Exponential(3,0,(1/Mu4)))

MIT24 LEAVE SYS4

OUT4 TERMINATE

GENERATE CHMOD

SAVEVALUE Imo1,V\$Imo1 ; модельна ймовірність обслуговування
; для системи 1

SAVEVALUE Imo2,V\$Imo2 ; модельна ймовірність обслуговування
; для системи 2

SAVEVALUE Imo3,V\$Imo3 ; модельна ймовірність обслуговування
; для системи 3

SAVEVALUE Imo4,V\$Imo4 ; модельна ймовірність обслуговування
; для системи 4

TERMINATE 1

У першому сегменті моделі забезпечується обчислення стаціонарних значень ймовірності обслуговування і середньої кількості зайнятих каналів для чотирьох різних розподілів інтервалу надходження замовлень і часу обслуговування згідно з формулами (5.18)-(5.21). Після цього вводяться арифметичні змінні Imo1-Imo4, за допомогою яких будуть підраховані модельні значення ймовірностей обслуговування для ОКП SYS1-SYS4.

У наступних чотирьох сегментах моделі здійснюється імітаційне моделювання одноканальних СМО з відмовами відповідно до чотирьох різних розподілів інтервалу надходження замовлень і часу обслуговування.

В останній частині моделі реалізується заданий час моделювання і модельні значення ймовірностей обслуговування заносяться до комірок зберігання.

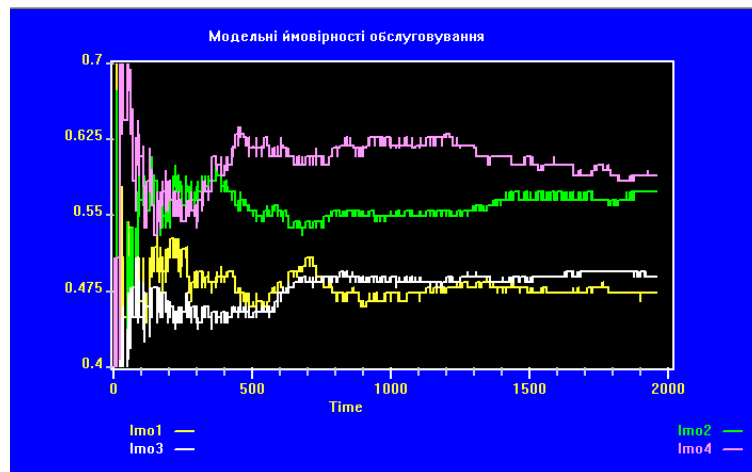


Рис. 5.15. Порівняння модельних значень імовірностей обслуговування для випадків 1-4

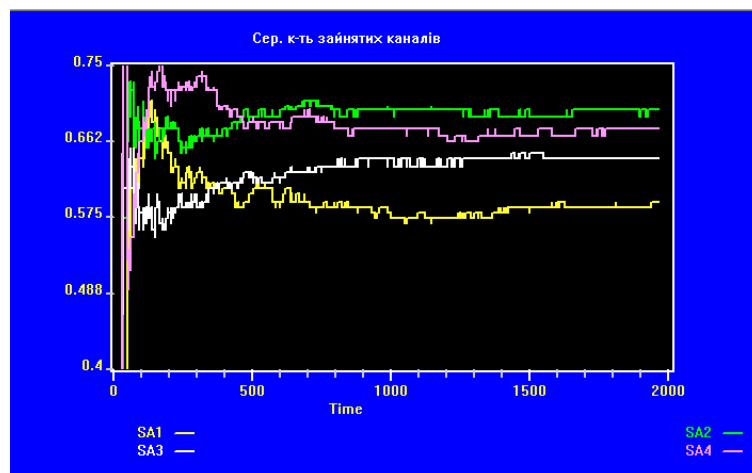


Рис. 5.16. Порівняння модельних значень середньої кількості зайнятих каналів для випадків 1-4

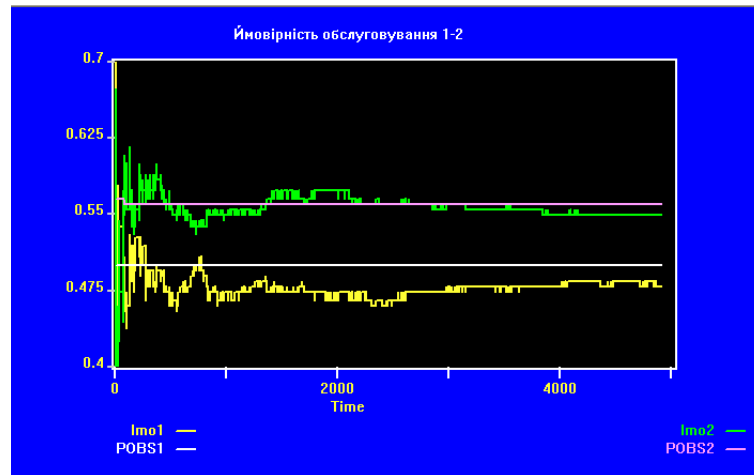


Рис. 5.17. Порівняння модельних і стаціонарних (POBS1, POBS2) значень ймовірностей обслуговування для випадків 1-2

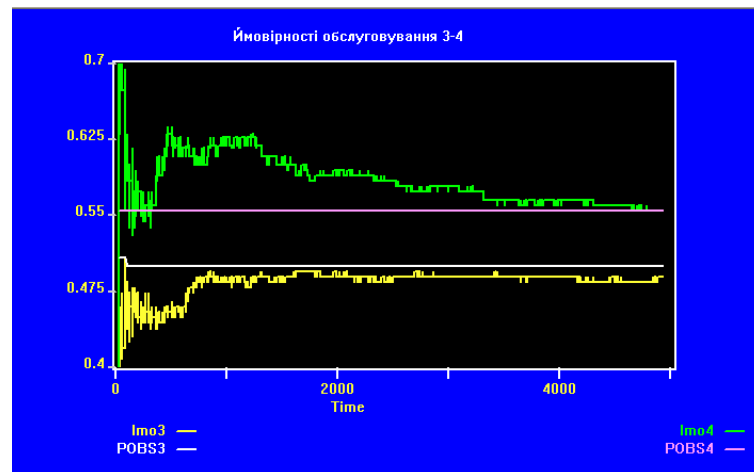


Рис. 5.18. Порівняння модельних і стаціонарних (POBS3, POBS4) значень ймовірностей обслуговування для випадків 3-4

Отримані графічні залежності та числові дані (рис. 5.15-5.18, табл. 5.2) показують, що для великих значень часу моделювання результати імітаційного моделювання у всіх чотирьох випадках збігаються з відповідними стаціонарними значеннями ймовірності обслуговування і середньої кількості зайнятих каналів, отриманими аналітичним методом. Результати моделювання для різних розподілів інтервалу надходження замовлень і часу обслуговування відрізняються між собою (рис. 5.15-5.16), хоч зі зростанням модельного часу відбувається їхнє зближення.

Таблиця 5.2

Час моделювання	Імовірність обслуговування (імітаційна модель)				Середня кількість зайнятих каналів (імітаційна модель)			
	1	2	3	4	1	2	3	4
100	0,478	0,560	0,440	0,625	0,664	0,645	0,569	0,680
1000	0,461	0,540	0,483	0,616	0,572	0,691	0,635	0,668
5000	0,475	0,542	0,482	0,550	0,621	0,699	0,616	0,690
10000	0,482	0,541	0,489	0,542	0,616	0,693	0,611	0,694
100000	0,496	0,557	0,498	0,552	0,616	0,695	0,625	0,688
500000	0,495	0,556	0,498	0,551	0,618	0,694	0,624	0,688
2000000	0,495	0,556	0,500	0,551	0,618	0,694	0,624	0,688
Стационарне значення	0,495	0,556	0,500	0,551	0,619	0,694	0,624	0,688

5.3. Моделювання обмежень на час перебування в черзі

Приклад 5.5. У триканальній СМО з очікуванням організувати дисципліну обслуговування, за якої замовлення не обслуговуються, якщо вони перебувають у черзі більше, ніж 1 од. модельного часу. Перевірити роботу моделі за допомогою таблиці розподілу часу перебування в черзі.

; Приклад 5.5.

```

La EQU 1/3          ; інтенсивність потоку замовлень
Mu EQU 1/10         ; інтенсивність потоку обслуговувань
WT QTABLE CHER,0,0.5,4 ; таблиця розподілу часу в черзі
Imo VARIABLE N$MIT2/N$VSI ; ймовірність обслуговування
SYS STORAGE 3      ; триканальна СМО
*****
GENERATE (Exponential(3,0,(1/La)))
ASSIGN Tobs,(Exponential(4,0,(1/Mu))) ; час обслуговування
VSI MARK TCH
QUEUE CHER
GATE SNF SYS          ; перевірка незаповненості БКП
TEST LE MP$TCH,1,MIT1 ; перевірка часу в черзі
ENTER SYS
DEPART CHER
ADVANCE P$Tobs
LEAVE SYS
MIT2 TERMINATE        ; обслужені замовлення
MIT1 DEPART CHER
TERMINATE             ; втрачені замовлення
*****
*****
GENERATE 5000          ; час моделювання
SAVEVALUE Imo,V$Imo
TERMINATE 1

```

Блок MARK, який розташований перед блоком QUEUE, записує в параметр з іменем TCH активного транзакта значення абсолютного модельного часу. Якщо БКП SYS незаповнений, то транзакт пройде через блок GATE і потрапить до блока TEST, де його час перебування в черзі (СЧА MP\$TCH) порівнюється з 1 (заданим обмеженням на час перебування в черзі). Якщо задана у блоці TEST умова виконується, то транзакт переходить до блока ENTER, а в іншому випадку він скеровується до блока MIT1, виходить з черги і знищується.

Проаналізуємо роботу моделі, скориставшись даними наведених нижче фрагментів стандартного звіту, отриманого для значення часу моделювання

5000 од. модельного часу, які містять інформацію про використані блоки і таблицю розподілу часу перебування в черзі.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1648	0	0
	2	ASSIGN	1648	0	0
VSI	3	MARK	1648	0	0
	4	QUEUE	1648	0	0
	5	GATE	1648	0	0
	6	TEST	1648	0	0
	7	ENTER	1075	0	0
	8	DEPART	1075	0	0
	9	ADVANCE	1075	1	0
	10	LEAVE	1074	0	0
MIT2	11	TERMINATE	1074	0	0
MIT1	12	DEPART	573	0	0
	13	TERMINATE	573	0	0

TABLE	MEAN	STD.DEV.	RANGE	RETRY FREQUENCY	CUM.%
WT	1.540	2.743		0	
			- 0.000	903	54.79
			0.000 - 0.500	80	59.65
			0.500 - 1.000	92	65.23
			1.000 -	573	100.00

За час моделювання було згенеровано 1648 транзактів, з них 1075 потрапили на обслуговування до БКП SYS, 1074 обслужено, 1 в момент виведення звіту перебував на обслуговуванні (у блоці ADVANCE) і 573 покинули чергу, не потрапивши на обслуговування, оскільки час перебування в черзі для них перевищував задане обмеження 1 од. модельного часу. Ймовірність обслуговування, яка для СМО з необмеженою довжиною черги, як відомо, дорівнює 1, у цьому випадку за рахунок обмеження часу перебування в черзі становить 0,652.

Кількість частотних інтервалів у таблиці WT розподілу часу перебування в черзі підібрано так, щоб до останнього частотного інтервалу потрапили лише ті транзакти, час перебування в черзі яких перевищує 1 од. модельного часу. Отже, кількість цих транзактів (573) збігається з даними, наведеними в першому фрагменті звіту. Середній час перебування в черзі становить 1,540, середнє квадратичне відхилення цього часу – 2,743.

Приклад 5.6. У триканальній СМО з очікуванням організувати дисципліну обслуговування, за якої замовлення не обслуговуються, якщо вони перебувають у черзі більше, ніж випадковий час ТРСН, розподілений за показниковим

законом з середнім значенням 1 од. модельного часу. Таку дисципліну обслуговування організувати двома способами: 1) з використанням блоків QUEUE і DEPART; 2) за допомогою списку користувача. Для генерування транзактів використати лише один блок GENERATE.

; Приклад 5.6.

```
La EQU 1/3 ; інтенсивність потоку замовлень
Mu EQU 1/10 ; інтенсивність потоку обслуговувань
Nu EQU 1 ; інтенсивність потоку виходів з черги
Per BVARIABLE (AC1-P1) 'LE' P$TPCH ; булева змінна
Imo1 VARIABLE N$MIT21/N$VSI1 ; ймовірність обслуговування 1
Imo2 VARIABLE N$MIT22/N$VSI2 ; ймовірність обслуговування 2
SYS1 STORAGE 3 ; триканальна СМО
SYS2 STORAGE 3 ; триканальна СМО
```

; Використання блоків QUEUE і

```
DEPART*****
    GENERATE (Exponential(3,0,(1/La)))
    ASSIGN TPCH,(Exponential(5,0,(1/Nu_))) ; запис
                                     ; обмеження на час у черзі
    ASSIGN Tobs,(Exponential(4,0,(1/Mu))) ; запис часу
                                     ; обслуговування
    SPLIT 1,VSI2 ; створити копію транзакта
VSI1 MARK TCHER
    QUEUE CHER
    GATE SNF SYS1 ; перевірка незаповненості БКП
TEST LE MP$TCHER,P$TPCH,MIT11 ; обмеження часу в черзі
    ENTER SYS1
    DEPART CHER
    ADVANCE P$Tobs
    LEAVE SYS1
MIT21 TERMINATE ; обслужені замовлення
MIT11 DEPART CHER
    TERMINATE ; втрачені замовлення
```

; Використання списку користувача

```
VSI2 GATE SNF SYS2,Sps ; перевірка незаповненості БКП
VhidENTER SYS2
    ADVANCE P$Tobs
MIT22 LEAVE SYS2
    UNLINK Spys,Vhid,1,BV$Per,,Term ; вивести зі списку
    TERMINATE ; обслужені з черги
```

```

Term TERMINATE           ; обслужені без черги
Sps ASSIGN 1,AC1
  LINK Spys,P1 ; (або FIFO) список користувача (втрачені
                ; замовлення)
*****
GENERATE 5000           ; час моделювання
SAVEVALUE lmo1,V$lmo1
SAVEVALUE lmo2,V$lmo2
TERMINATE 1
    
```

Для аналізу роботи моделі використаємо фрагменти стандартного звіту, отриманого для значення часу моделювання 5000 од. модельного часу, які містять інформацію про використані блоки та про об'єкт "черга".

Після блока GENERATE розташовані два блоки ASSIGN. Перший з цих блоків здійснює запис у параметр TPCN активного транзакта випадкового часу, який задає обмеження на час перебування в черзі, а другий блок у параметр Tobs записує випадковий час обслуговування. Після цього транзакт потрапляє до блока SPLIT, призначеного для створення в моделі заданої кількості копій транзактів.

Блок SPLIT має такий формат:

```
SPLIT  A,[B],[C]
```

Значення операндів: A – кількість копій транзактів; B – ім'я або номер блока, до якого переходять копії транзакта-оригінала. Операндом C можна задавати номер параметра, який слугує для присвоєння копіям послідовних номерів. Після проходження блока SPLIT транзакт-оригінал переходить до наступного блока, а всі копії скеровуються за адресою, вказаною операндом B, або до наступного блока, якщо операнда B немає.

У моделі, яку ми розглядаємо, блок SPLIT слугує для створення однакових умов для порівняння результатів роботи двох сегментів моделі, які задають одну й ту ж саму дисципліну обслуговування різними способами. Цей блок створює одну копію згенерованого блоком GENERATE транзакта, яка переходить до блока з міткою VSI2.

	LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
		1	GENERATE		1648		0		0
		2	ASSIGN		1648		0		0
		3	ASSIGN		1648		0		0
		4	SPLIT		1648		0		0
VSI1		5	MARK		1648		0		0
		6	QUEUE		1648		0		0
		7	GATE		1648		0		0
		8	TEST		1648		0		0
		9	ENTER		1042		0		0

5. МОДЕЛІ РОЗІМКНЕНИХ СИСТЕМ МАСОВОГО
ОБСЛУГОВУВАННЯ

	10	DEPART	1042	0	0
	11	ADVANCE	1042	1	0
	12	LEAVE	1041	0	0
MIT21	13	TERMINATE	1041	0	0
MIT11	14	DEPART	606	0	0
	15	TERMINATE	606	0	0
VSI2	16	GATE	1648	0	0
VHID	17	ENTER	1042	0	0
	18	ADVANCE	1042	1	0
MIT22	19	LEAVE	1041	0	0
	20	UNLINK	1041	0	0
	21	TERMINATE	130	0	0
TERM	22	TERMINATE	911	0	0
SPS	23	ASSIGN	736	0	0
	24	LINK	736	606	0

	QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.	CONT.	AVE.	TIME	AVE.(-0)
CHER	13	0	1648	912	0.627	1.902	4.258			

Сегмент моделі, в якому використані блоки QUEUE і DEPART, працює так само, як в моделі прикладу 5.5. Згідно зі стандартним звітом, за час моделювання було згенеровано 1648 транзактів, з них 1042 потрапили на обслуговування в БКП SYS1, 1041 обслужено, 1 у момент виведення звіту перебував на обслуговуванні (у блоці ADVANCE) і 606 покинули чергу, не потрапивши на обслуговування, оскільки час перебування в черзі для них перевищував задане випадкове обмеження TRCH.

Робота сегмента моделі, в якому використано список користувача, починається з потраплянням транзакта-копії від блока SPLIT до блока GATE. Якщо БКП SYS2 незаповнений, то цей транзакт пройде через блок GATE і потрапить до БКП SYS2 (таких транзактів згідно з даними стандартного звіту є 1042, як і для БКП SYS1). Якщо ж у момент надходження транзакта БКП SYS2 заповнений, то блок GATE скеровує транзакт до блока ASSIGN з міткою Sps (таких транзактів – 736, стільки ж, як і транзактів, які перебували в черзі до БКП SYS1 після вилучення „нульових” входжень до неї згідно з інформацією про блок QUEUE CHER у стандартному звіті). У блоці ASSIGN в параметр P1 транзакта записується абсолютний модельний час, який відповідає моментові введення його до списку користувача Sps у наступному після блока ASSIGN блоці LINK.

У полі операнда D блока UNLINK використано булеву змінну Per, яка дає змогу порівнювати час перебування транзакта в черзі із заданим обмеженням TRCH. Час перебування в черзі визначено як різницю абсолютного модельного часу в момент входження активного транзакта до блока UNLINK і абсолютного модельного часу в момент введення транзакта до списку користувача. Якщо

результат обчислення булевої змінної ненульовий, тобто задана в булевій змінній умова виконується, то зі списку користувача виводиться *один* транзакт (відповідно до значення операнда С блока UNLINK) і скеровується до мітки Vhid на обслуговування в БКП SYS2.

Операнд F блока UNLINK вказує мітку TERM блока TERMINATE, куди переходить транзакт, який виходить з блока UNLINK, якщо зі списку користувача не виведено жодного транзакта (таких транзактів є 911). Порівняння з даними нульових входжень у чергу QUEUE CHER у стандартному звіті показує, що це – кількість замовлень, обслужених без черги. Останнє замовлення, яке потрапило на обслуговування без черги, в момент завершення роботи моделі перебувало на обслуговуванні. Замовлень, які потрапили на обслуговування після перебування в черзі (у списку користувача), є 130 – за кількістю входжень до блока TERMINATE, розташованого після блока UNLINK. Отже, в БКП SYS2, як і в БКП SYS1, всього обслужено 1041 замовлень. Усі необслужені замовлення (їх 606) залишилися у блоці LINK у списку користувача. Ймовірності обслуговування для БКП SYS1 і БКП SYS2, які виводяться у стандартному звіті в комірках зберігання Imo1 та Imo2 відповідно, однакові й дорівнюють 0,632.

Отже, застосовані нами два різні способи організації дисципліни обслуговування дають однакові результати.

5.4. Моделювання обслуговування неоднорідних замовлень
та пріоритетного обслуговування

```
; Приклад 5.7.
; задання вхідних даних
LamEQU 1/2          ; інтенсивність потоку замовлень
Mu1EQU 1/10
Mu2EQU 1/20
Mu3EQU 1/30
Mu4EQU 1/40
ALF1 EQU 0.1
ALF2 EQU 0.2
ALF3 EQU 0.3
ALF4 EQU 0.4
Mu EQU 1/(ALF1/Mu1+ALF2/Mu2+ALF3/Mu3+ALF4/Mu4)
; (інтенсивність потоку обслуговувань з урахуванням неоднорідності)
Typ FUNCTION RN15,D4          ; тип замовлення
.1,1/.3,2/.6,3/1,4
Mean FUNCTION P$TypZ,L4      ; середній час обслуговування
1,10/2,20/3,30/4,40
Imo VARIABLE N$MIT2/N$MIT1 ; ймовірність обслуговування
SYS STORAGE 3                ; 3-канальна СМО
CHMOD EQU 30                 ; час моделювання
*****

; Інтегрування системи диф. рівнянь для ймовірностей станів
p0_ EQU 1                    ; початкові умови
p1_ EQU 0
p2_ EQU 0
p3_ EQU 0
p0_ INTEGRATE (-Lam#p0_+Mu#p1_)
p1_ INTEGRATE (-(Lam+Mu)#p1_+Lam#p0_+2#Mu#p2_)
p2_ INTEGRATE (-(Lam+2#Mu)#p2_+Lam#p1_+3#Mu#p3_)
p3_ INTEGRATE (-3#Mu#p3_+Lam#p2_)
*****

; Імітаційна модель СМО
GENERATE (Exponential(5,0,(1/Lam)))
MIT1 GATE SNF SYS,OUT ; перевірка умови незаповненості БКП
ASSIGN TypZ, FN$Typ
ENTER SYS
ADVANCE (Exponential(3,0, FN$Mean))
MIT2 LEAVE SYS
OUT TERMINATE
GENERATE CHMOD

; Порівняння результатів *****
SAVEVALUE Imo,V$Imo          ; ймовірність обслуговування
; (імітаційна модель)
```

```
SAVEVALUE TImo,(1-p3_) ; імовірність обслуговування
                        ; (аналітична модель)
; Середня кількість зайнятих каналів (аналітична модель)
SAVEVALUE SKZK,(p1_+2#p2_+3#p3_)
TERMINATE 1
```

Ми розглядаємо найпростіший потік замовлень з інтенсивністю $\lambda=1/2$. Час обслуговування розподілений за показниковим законом.

Система рівнянь для ймовірностей станів записана згідно з (5.23) ($n=3$) й інтегрується за допомогою операторів INTEGRATE.

Наявність вільних каналів у момент надходження замовлення перевіряється блоком GATE. Якщо вільних каналів немає, то транзакт скеровується до блоку TERMINATE з міткою OUT для знищення.

Функція Тур у моделі задає тип замовлення, а функція Mean – середній час обслуговування замовлення залежно від його типу. У блоці ASSIGN у параметр ТурZ транзактів записується тип замовлення, який розігрується за допомогою функції Тур. Увійшовши до блоку ADVANCE, транзакт затримується на випадковий час, який визначається функцією Mean відповідно до типу замовлення, який записаний у параметрі ТурZ цього транзакта.

В останній частині моделі реалізується заданий час моделювання, а в комірки зберігання заносяться модельне та аналітичне значення ймовірності обслуговування й аналітичне значення середньої кількості зайнятих каналів.

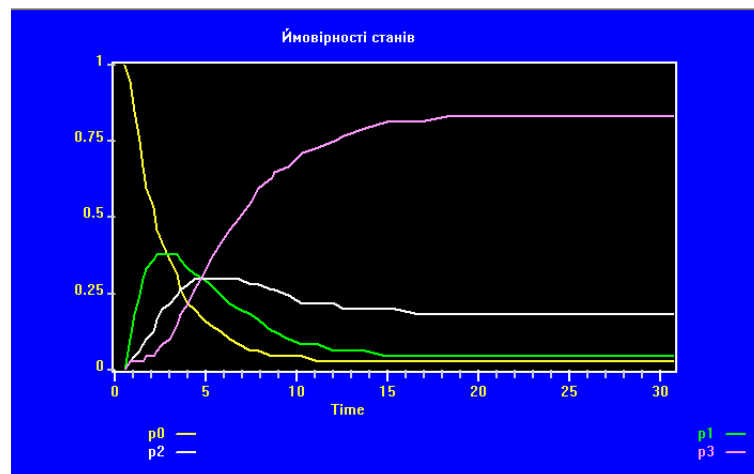


Рис. 5.19. Залежності ймовірностей станів від часу, отримані числовим інтегруванням рівнянь аналітичної моделі

Графічні залежності ймовірностей станів системи від часу, отримані засобами GPSS World (у вікні Plot) за результатами числового інтегрування системи чотирьох диференціальних рівнянь для значення часу моделювання 30, зображено на рис. 5.19.

Графік показує, що на проміжку модельного часу (15;20) згідно з

5.4. Моделювання обслуговування неоднорідних замовлень та пріоритетного обслуговування

результатами аналітичного моделювання має досягатися стаціонарний режим роботи системи.

Ймовірність обслуговування в імітаційній моделі обчислено як відношення кількості обслужених замовлень (транзактів, які пройшли через блок з міткою МІТ2) до загальної кількості замовлень (транзактів, які пройшли через блок з міткою МІТ1). В аналітичній моделі ймовірність обслуговування підраховуємо як різницю $1-p3_$, де $p3_$ – розв’язок системи диференціальних рівнянь для ймовірностей станів, отриманий шляхом числового інтегрування.

Динаміку зближення результатів імітаційного та аналітичного моделювання зі зростанням часу моделювання наведено у табл. 5.3.

Таблиця 5.3

Час моделювання	Ймовірність обслуговування (імітаційна модель)	Ймовірність обслуговування (аналітична модель)	Середня кількість зайнятих каналів (імітаційна модель)	Середня кількість зайнятих каналів (аналітична модель)
10	0,000	0,289	0,323	2,614
100	0,245	0,186	2,475	2,789
1000	0,163	0,186	2,790	2,789
5000	0,188	0,186	2,769	2,789
10000	0,188	0,186	2,779	2,789
100000	0,183	0,186	2,789	2,789
1000000	0,186	0,186	2,788	2,789
2000000	0,186	0,186	2,789	2,789

Приклад 5.8. Побудувати модель триканальної СМО з обмеженням на довжину черги та з неоднорідними замовленнями чотирьох типів. Потік замовлень найпростіший з інтенсивністю $\lambda=1/2$. Ймовірності надходження замовлень i -го типу $\alpha_i=0,1i$ ($i=\overline{1,4}$) відповідно, а середній час обслуговування відповідно становить 10, 20, 30 і 40 хв. Час обслуговування розподілений за показниковим законом. Передбачити:

- пріоритетне обслуговування замовлень за принципом: чим менший середній час обслуговування, тим вищий пріоритет замовлення;
- обчислення середнього часу очікування в черзі та ймовірностей обслуговування для замовлень кожного типу.

Порівняти результати моделювання для випадків:

- пріоритетного обслуговування за принципом 1: чим менший середній час обслуговування, тим вищий пріоритет замовлення;
- пріоритетного обслуговування за протилежним принципом (принципом 2);
- безпріоритетного обслуговування.

; Приклад 5.8.

; задання вхідних даних

```

SYS STORAGE 3 ; 3-канальна СМО
CHMOD EQU 10000 ; час моделювання
LamEQU 1/2 ; інтенсивність потоку замовлень
Typ FUNCTION RN15,D4 ; тип замовлення
.1,1/.3,2/.6,3/1,4
Mean FUNCTION P1,D4 ; середній час обслуговування
1,10/2,20/3,30/4,40
Dov FUNCTION P1,D4 ; довжини черг за типами замовлень=2
1,2/2,2/3,2/4,2
Clas FUNCTION P1,D4 ; пріоритет замовлення
1,4/2,3/3,2/4,1
; визначення ймовірностей обслуговування замовлень
Imo VARIABLE N$Obs/N$Vhid ; будь-якого типу
Imo1 VARIABLE N$Term1/X1 ; 1-го типу
Imo2 VARIABLE N$Term2/X2 ; 2-го типу
Imo3 VARIABLE N$Term3/X3 ; 3-го типу
Imo4 VARIABLE N$Term4/X4 ; 4-го типу
*****
; Імітаційна модель СМО
GENERATE (Exponential(5,0,(1/Lam)))
VhidASSIGN 1, FN$Typ ; розігрування типу замовлення
PRIORITY FN$Clas ; визначення пріоритету замовлення
SAVEVALUE P1+,1 ; облік замовлень за типами
; визначення типів замовлень і перевірка довжин черг
TEST E P1,1,Mit1 ; 1-го типу?
TEST L Q1, FN$Dov,OUT ; чи є місце у черзі?
TRANSFER ,Mit4 ; скерування в чергу
Mit1 TEST E P1,2,Mit2 ; 2-го типу?
TEST L Q2, FN$Dov,OUT ; чи є місце в черзі?
TRANSFER ,Mit4 ; скерування в чергу
Mit2 TEST E P1,3,Mit3 ; 3-го типу?
TEST L Q3, FN$Dov,OUT ; чи є місце в черзі?
TRANSFER ,Mit4 ; скерування в чергу
Mit3 TEST L Q4, FN$Dov,OUT ; чи є місце в черзі?
Mit4 QUEUE SPCHE ; спільна черга
QUEUE P1 ; черга за типом замовлення
ENTER SYS ; зайняти вільний канал
DEPART SPCHE ; покинути спільну чергу
DEPART P1 ; покинути чергу за типом замовлень
ADVANCE (Exponential(3,0, FN$Mean))
LEAVE SYS ; звільнити канал обслуговування
; облік обслужених замовлень і видалення замовлень з моделі
Obs TEST E P1,1,Typ2 ; 1-го типу?

```

Term1 TERMINATE ; вилучення обслужених замовлень 1-го типу
Typ2 TEST E P1,2,Typ3 ; 2-го типу?
Term2 TERMINATE ; вилучення обслужених замовлень 2-го типу
Typ3 TEST E P1,3,Term4 ; 3-го типу?
Term3 TERMINATE ; вилучення обслужених замовлень 3-го типу
Term4 TERMINATE ; вилучення обслужених замовлень 4-го типу
OUT TERMINATE ; вилучення втрачених замовлень
; Сегмент задання часу моделювання та розрахунку результатів
GENERATE CHMOD
; розрахунок ймовірностей обслуговування замовлень
SAVEVALUE Imo,V\$Imo ; будь-якого типу
SAVEVALUE Imo1,V\$Imo1 ; 1-го типу
SAVEVALUE Imo2,V\$Imo2 ; 2-го типу
SAVEVALUE Imo3,V\$Imo3 ; 3-го типу
SAVEVALUE Imo4,V\$Imo4 ; 4-го типу
TERMINATE 1

У моделі функція Typ задає тип замовлення, а функції Mean, Dov і Clas відповідно – середній час обслуговування, довжину черги та пріоритетний клас замовлення залежно від його типу. Блок ASSIGN у перший параметр транзактів записує тип замовлення, розіграний за допомогою функції Typ, а блок PRIORITY активному транзактові присвоює пріоритет, який відповідає типові замовлення. Кожне входження транзакта до блоку SAVEVALUE приводить до збільшення на 1 вмісту комірки з номером, який записаний у першому параметрі транзакта.

Далі в моделі розташовано кілька пар блоків TEST. Перший блок кожної пари визначає тип замовлення, а другий – перевіряє наявність вільного місця в черзі, яка відповідає типові замовлення. Якщо місце у черзі немає, то транзакт скеровується до блоку TERMINATE з міткою OUT для знищення. В іншому випадку транзакт переходить до спільної черги, а потім до черги за типом замовлення. Отже, в моделі створено п'ять черг: одна спільна з іменем SPCH та чотири з номерами 1, 2, 3 і 4. Завдяки непрямій адресації останні чотири черги створені за допомогою однієї пари блоків QUEUE-DEPART.

Для забезпечення можливості обчислення ймовірностей обслуговування за типами замовлень вилучення обслужених замовлень з моделі здійснено також за типами замовлень з використанням блоків TEST.

Для отримання результатів моделювання у випадку пріоритетного обслуговування за принципом 2: чим більший середній час обслуговування, тим вищий пріоритет замовлення, потрібно функцію Clas задати у такому вигляді

Clas FUNCTION P1,D4 ; пріоритет замовлення
1,1/2,2/3,3/4,4

Модель працюватиме у безпріоритетному режимі обслуговування, якщо вилучити блок PRIORITY.

Нижче наведено фрагменти стандартного звіту, отримані для кожного з трьох розглянутих режимів обслуговування для значення часу моделювання 10000.

Пріоритетне обслуговування за принципом 1

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)
1	2	0	468	1	0.377	8.063	8.080
2	2	0	637	1	1.013	15.899	15.924
3	2	2	353	3	1.677	47.496	47.903
4	2	2	53	1	1.963	370.284	377.405
SPCHE	8	4	1511	6	5.029	33.284	33.417

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	DELAY
SYS	3	0	0	3	1507	1	2.996	0.999	4

SAVEVALUE	RETRY	VALUE
1	0	504.000
2	0	1013.000
3	0	1478.000
4	0	2028.000
IMO	0	0.299
IMO1	0	0.927
IMO2	0	0.628
IMO3	0	0.237
IMO4	0	0.025

Пріоритетне обслуговування за принципом 2

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)
1	2	2	2	0	1.990	9952.492	9952.492
2	2	2	6	1	1.994	3323.757	3988.508
3	2	2	84	1	1.920	228.608	231.362
4	2	0	697	1	1.534	22.007	22.039
SPCHE	8	6	789	3	7.439	94.283	94.643

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	DELAY
SYS	3	0	0	3	783	1	2.997	0.999	6

SAVEVALUE	RETRY	VALUE
1	0	504.000

5.4. Моделювання обслуговування неоднорідних замовлень та пріоритетного обслуговування

2	0	1013.000
3	0	1478.000
4	0	2028.000
IMO	0	0.155
IMO1	0	0
IMO2	0	0.004
IMO3	0	0.055
IMO4	0	0.342

Безпріоритетне обслуговування

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)
1	2	0	227	0	1.405	61.891	61.891
2	2	2	284	1	1.656	58.317	58.523
3	2	1	307	1	1.751	57.027	57.213
4	2	1	324	1	1.811	55.902	56.075
SPCHE	8	4	1142	3	6.623	57.995	58.148

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	DELAY
SYS	3	0	0	3	1138	1	2.997	0.999	4

SAVEVALUE	RETRY	VALUE
1	0	504.000
2	0	1013.000
3	0	1478.000
4	0	2028.000
IMO	0	0.226
IMO1	0	0.450
IMO2	0	0.278
IMO3	0	0.206
IMO4	0	0.158

За час моделювання в кожному з трьох випадків до СМО надійшло 504 замовлення 1-го типу, 1013 – 2-го типу, 1478 – 3-го типу і 2028 замовлень 4-го типу. З них на обслуговування потрапили 1507 замовлень у випадку пріоритетного обслуговування за принципом 1, 1138 замовлень у випадку безпріоритетного обслуговування і 783 замовлення у випадку пріоритетного обслуговування за принципом 2. Імовірності обслуговування замовлення незалежно від його типу в цих випадках відповідно дорівнюють 0,299; 0,226 і 0,155. Імовірності обслуговування за типами замовлень збільшуються зі зростанням пріоритету замовлення, а у випадку безпріоритетного обслуговування – зменшуються зі збільшенням номера типу.

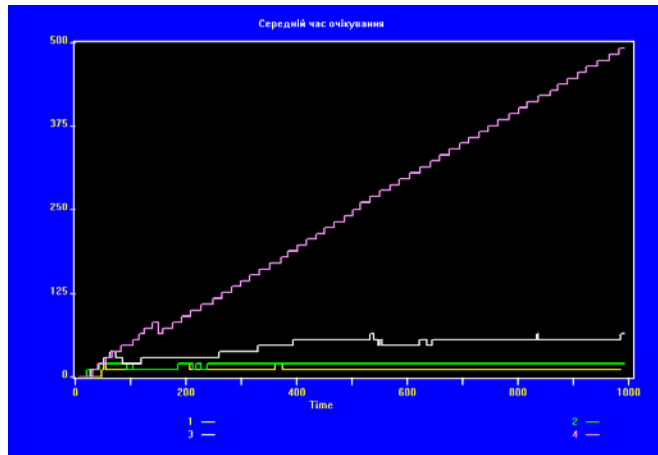


Рис. 5.20. Залежності середнього часу очікування від часу для замовлень i -го типу ($i = \overline{1,4}$) у режимі пріоритетного обслуговування за принципом 1

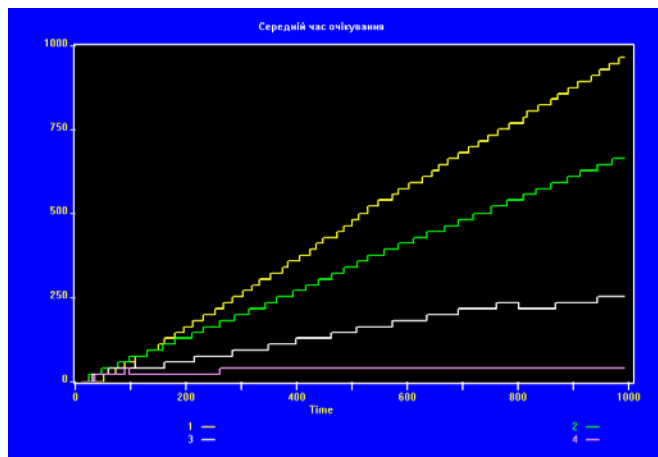


Рис. 5.21. Залежності середнього часу очікування від часу для замовлень i -го типу ($i = \overline{1,4}$) у режимі пріоритетного обслуговування за принципом 2

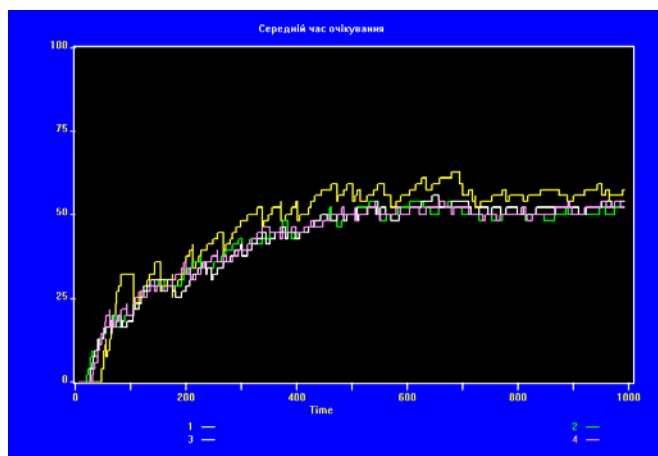


Рис. 5.22. Залежності середнього часу очікування від часу для замовлень i -го типу ($i = \overline{1,4}$) у режимі безпріоритетного обслуговування

У випадку безпріоритетного обслуговування середній час очікування для замовлень усіх типів приблизно однаковий і стабілізується зі зростанням модельного часу (рис. 5.22). У випадках пріоритетного обслуговування (рис. 5.20-5.21) середній час очікування найменший і теж стабілізується зі зростанням модельного часу для замовлень з найвищим пріоритетом. Для замовлень з низьким пріоритетом середній час очікування більший і різко зростає зі зростанням модельного часу.

5.5. Моделювання збору статистики з кожного каналу обслуговування окремо

Приклад 5.9. Побудувати модель чотириканальної СМО з двома місцями у черзі, передбачивши можливість отримання статистики обслуговування з кожного каналу обслуговування.

```

; Приклад 5.9.
; задання вхідних даних
Lam    EQU 1/2    ; інтенсивність потоку замовлень
Mu     EQU 1/10   ; інтенсивність потоку обслуговувань
Em     EQU 2      ; кількість місць у черзі
Imo    VARIABLE N$MIT2/N$MIT1 ; ймовірність обслуговування
CHMOD  EQU 1000  ; час моделювання
*****
; Імітаційна модель СМО
GENERATE (Exponential(1,0,(1/Lam)))
MIT1    TEST L Q$CHER,Em,OUT ; чи є вільне місце в черзі
QUEUE  CHER
TRANSFER ALL,KAN1,KAN4,3 ; знайти вільний канал
KAN1 SEIZE CAN1    ; чи вільний канал 1?
ASSIGN 1,CAN1    ; записати номер каналу
TRANSFER,VYH    ; на вихід з черги
SEIZE CAN2      ; чи вільний канал 2?
ASSIGN 1,CAN2    ; записати номер каналу
TRANSFER,VYH    ; на вихід з черги
SEIZE CAN3      ; чи вільний канал 3?
ASSIGN 1,CAN3    ; записати номер каналу
TRANSFER,VYH    ; на вихід з черги
KAN4 SEIZE CAN4    ; чи вільний канал 4?
ASSIGN 1,CAN4    ; записати номер каналу
VYH    DEPART CHER ; вийти з черги
ADVANCE (Exponential(1,0,(1/Mu))) ; обслуговування
MIT2    RELEASE P1    ; покинути канал
OUT     TERMINATE    ; вилучення замовлень
GENERATE CHMOD ; реалізувати час моделювання
SAVEVALUE Imo,V$Imo ; запис імовірності обслуговування
TERMINATE 1

```

Ми розглядаємо СМО з найпростішими потоками замовлень і обслуговувань з інтенсивностями $\lambda=1/2$ і $\mu=1/10$ відповідно. Наявність вільних місць у черзі в момент надходження замовлення перевіряє блок TEST. Якщо вільних місць немає, то транзакт скеровується до блока TERMINATE з міткою

OUT для знищення.

Якщо є вільне місце в черзі, то активний транзакт потрапляє до блока TRANSFER із записом ALL у полі операнда A, за допомогою якого шукає вільний канал обслуговування. Дочекавшись звільнення каналу, транзакт потрапляє до блока SEIZE, а потім до блока ASSIGN. Цей блок у перший параметр транзакта записує ім'я каналу, в якому буде здійснюватися обслуговування. Завдяки цьому замість чотирьох блоків RELEASE для чотирьох каналів використано один такий блок, в якому у полі операнда A за допомогою непрямой адресації зазначене ім'я каналу, записане в перший параметр транзакта. Перед входженням до блока RELEASE транзакт покидає чергу і затримується на обслуговування у блоці ADVANCE.

Нижче наведено фрагменти стандартного звіту для цієї моделі та з метою порівняння для моделі прикладу 5.2, отримані для значення часу моделювання 1000. Стандартний звіт для моделі прикладу 5.2 містить інформацію про весь чотириканальний пристрій, а звіт даної моделі – інформацію про кожен канал цього пристрою. Якщо просумувати такі показники, як кількість входжень до каналу обслуговування та коефіцієнт використання каналу, отримані для кожного каналу окремо, то одержимо результат, який збігається з відповідним показником (сумарна кількість входжень і середня кількість зайнятих каналів) чотириканального пристрою загалом.

Фрагмент стандартного звіту для моделі прикладу 5.9

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER
CAN1	90	0.888	9.868	1	477	0	0
CAN2	86	0.912	10.600	1	487	0	0
CAN3	89	0.908	10.197	1	468	0	0
CAN4	81	0.857	10.577	1	476	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)
CHER	2	0	346	159	0.769	2.223	4.112

SAVEVALUE	RETRY	VALUE
IMO	0	0.692

Фрагмент стандартного звіту для моделі прикладу 5.2

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SYS	4	0	0	4	346	1	3.564	0.891

5.6. Побудова моделей систем з обмеженнями на коефіцієнти використання каналів

Приклад 5.10. Побудувати модель триканальної СМО з відмовами і спеціальним розподілом замовлень між каналами обслуговування залежно від значень їхніх коефіцієнтів використання (КВ). Якщо для першого каналу КВ перевищує значення 0,7, для другого – 0,8, то замовлення скеровується у третій канал. У свою чергу, якщо для третього каналу КВ перевищує значення 0,9, то замовлення не обслуговуються і втрачаються.

; Приклад 5.10.

; Задання булевих змінних

Per1 BVARIABLE F1'AND'F2'AND'F3

Per2 BVARIABLE F1'OR'(FR1>700)

Per3 BVARIABLE F2'OR'(FR2>800)

Per4 BVARIABLE F3'OR'(FR3>900)

; Сегмент імітування надходження і опрацювання замовлень

GENERATE (Exponential(5,0,6))

TEST E BV\$Per1,0,MIT1 ; Чи зайняті всі 3 канали?

TEST E BV\$Per2,0,MIT2 ; Чи зайнятий 1 канал або КВ>0,7?

; Імітування роботи 1-го каналу

SEIZE 1

ADVANCE (Uniform(8,10,60))

RELEASE 1

TERMINATE

; Імітування роботи 2-го каналу

MIT2 TEST E BV\$Per3,0,MIT3 ; Чи зайнятий 2 канал або КВ>0,8?

SEIZE 2

ADVANCE (Uniform(8,15,65))

RELEASE 2

TERMINATE

; Імітування роботи 3-го каналу

MIT3 TEST E BV\$Per4,0,MIT1 ; Чи зайнятий 3 канал або КВ>0,9?

SEIZE 3

ADVANCE (Uniform(8,5,195))

RELEASE 3

TERMINATE

MIT1 TERMINATE

; Сегмент задання часу моделювання

GENERATE 10000

TERMINATE 1

Булева змінна $Per1$ набуває значення 1 у випадку, коли ОКП1, ОКП2 і ОКП3 зайняті. У цьому випадку умова в першому блоці TEST не виконується і транзакт скеровується до блока TERMINATE з міткою MIT1 для знищення.

У другому блоці TEST порівнюється значення булевої змінної $Per2$ з нулем. Задана умова не виконується, якщо булева змінна набуває значення 1, тобто ОКП1 зайнятий або його коефіцієнт використання перевищує значення 0,7. (Коефіцієнт використання в булевій змінній записують у тисячних частках як ціле число). Тоді транзакт скеровується до ОКП2, де у блоці TEST з міткою MIT2 здійснюється така сама перевірка, як і в попередньому випадку, з тою лише відмінністю, що використано булеву змінну $Per3$. У випадку невиконання умови (значення булевої змінної рівне 1), транзакт скеровується до ОКП3 (до блока TEST з міткою MIT3).

Якщо ОКП3 вільний і його коефіцієнт використання не перевищує 0,9 (значення булевої змінної $Per4$ рівне 0), умова у блоці TEST виконується і транзакт скеровується на обслуговування. В іншому випадку транзакт скеровується до блока TERMINATE з міткою MIT1 для знищення.

Наведемо фрагмент стандартного звіту, одержаного для значення часу моделювання 10000.

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER
1	210	0.701	33.364	1	0	0	0
2	196	0.801	40.842	1	0	0	0
3	90	0.900	99.987	1	0	0	0

Отже, коефіцієнти використання є такими, як зазначено в умові прикладу.

5.8. Багатофазні системи

Аналітичний аналіз багатофазних систем у загальному випадку проблематичний, оскільки вхідний потік кожної наступної фази є вихідним потоком попередньої і загалом має післядію. Однак, якщо на вхід СМО з необмеженою чергою надходить найпростіший потік замовлень, а час обслуговування показниковий, то вихідний потік цієї СМО – найпростіший, з тою самою інтенсивністю λ , що й вхідний. Тому багатофазну СМО з необмеженою чергою перед кожною фазою, найпростішим вхідним потоком замовлень і показниковим законом обслуговування на кожній фазі, можна аналізувати як послідовність найпростіших СМО.

Приклад 5.15. Система масового обслуговування – взуттєвий магазин, у якому кожен покупець проходить три фази обслуговування:

- вибір взуття;
- сплата грошей у касу;
- отримання покупки на контролі.

У магазин прибуває найпростіший потік покупців з інтенсивністю $\lambda=45$ осіб/год.

У відділі вибору взуття є чотири стільці, на яких, покупці можуть самостійно вибирати і міряти взуття. Середній час вибору взуття 5 хв. Вибравши взуття, покупець іде до каси, де знову стає в чергу (каса у магазині одна). Середній час оплати товару в касі 1 хв. Розрахувавшись у касі, покупець іде на контроль, де стає у нову чергу та отримує покупку. На контролі працюють три продавці; середній час отримання покупки 2 хв. Усі потоки подій – найпростіші.

Розглядаючи магазин як трифазну СМО, за допомогою імітаційної моделі знайти характеристики її ефективності:

- середню кількість покупців у чергах до кожної з трьох фаз обслуговування;
- середню кількість покупців, які перебувають на кожній фазі обслуговування;
- середній час очікування покупця в черзі до кожної фази;
- середній час перебування покупця в кожній фазі обслуговування;
- загальний середній час, який витрачає покупець на придбання взуття в магазині.

Результати порівняти з отриманими в [2] аналітичним методом.

; Приклад 5.15.

; задання вхідних даних

SNPER TABLE M1,0,10,20

```
*****
PHASE1 STORAGE 4
PHASE2 STORAGE 1
PHASE3 STORAGE 3
*****
      GENERATE (Exponential(1,0,(4/3)))
*****
;   перша фаза обслуговування
*****
      QUEUE 1
      ENTER PHASE1
      DEPART 1
      ADVANCE (Exponential(1,0,5))
      LEAVE PHASE1
*****
;   друга фаза обслуговування
*****
      QUEUE 2
      ENTER PHASE2
      DEPART 2
      ADVANCE (Exponential(1,0,1))
      LEAVE PHASE2
*****
;   третя фаза обслуговування
*****
      QUEUE 3
      ENTER PHASE3
      DEPART 3
      ADVANCE (Exponential(1,0,2))
      LEAVE PHASE3
      TERMINATE
*****
      GENERATE 200000 ; реалізувати час моделювання
*****
;   обчислення характеристик ефективності трифазної СМО
*****
;   середня кількість покупців на 1-ій, 2-ій, 3-ій фазах обслуговування
*****
      SAVEVALUE SKP1,(SA$PHASE1+QA1)
      SAVEVALUE SKP2,(SA$PHASE2+QA2)
      SAVEVALUE SKP3,(SA$PHASE3+QA3)
*****
;   середній час обслуговування покупця в кожній фазі
```

```
SAVEVALUE CHOBS1,ST$PHASE1
SAVEVALUE CHOBS2,ST$PHASE2
SAVEVALUE CHOBS3,ST$PHASE3
```

; середній час перебування покупця в кожній фазі обслуговування

```
SAVEVALUE PER1,(ST$PHASE1+QT1)
SAVEVALUE PER2,(ST$PHASE2+QT2)
SAVEVALUE PER3,(ST$PHASE3+QT3)
```

; загальний середній час перебування покупця в магазині

```
SAVEVALUE CHPER,(X$PER1+X$PER2+X$PER3)
```

```
TERMINATE 1
START 1
```

За одиницю модельного часу в моделі прийнято 1 хв. Тому інтенсивність потоку замовлень $\lambda=0.75$, а $1/\lambda=4/3$.

Інформацію про середню кількість покупців у чергах до кожної з трьох фаз обслуговування та середній час очікування покупця в черзі до кожної фази можна знайти у стандартному звіті моделі серед характеристик об'єктів QUEUE 1, QUEUE 2 і QUEUE 3 відповідно. Інші характеристики обчислені за допомогою блоків SAVEVALUE та СЧА:

- SA – середня кількість зайнятих каналів;
- QA – середня довжина черги;
- ST – середній час обслуговування одним каналом одного замовлення;
- QT – середній час перебування в черзі;
- X – вміст комірки.

Результати обчислень для значення часу моделювання 2000000 зведено у табл. 5.8.

Таблиця 5.8

Номер фази	Середня кількість покупців у чергах до кожної з фаз	
	Аналітичний метод згідно з [2]	Імітаційна модель
1	13,01	12,53
2	2,25	2,25
3	0,24	0,24
Номер фази	Середня кількість покупців на кожній фазі	
	Аналітичний метод згідно з [2]	Імітаційна модель
1	16,76	16,28
2	3,00	3,00
3	1,74	1,74
Номер фази	Середній час очікування в черзі до кожної фази, хв	
	Аналітичний метод згідно з [2]	Імітаційна модель
1	17,3	16,71
2	3,00	3,00
3	0,32	0,32
Номер фази	Середній час перебування в кожній фазі, хв	
	Аналітичний метод згідно з [2]	Імітаційна модель
1	22,32	21,71
2	4,00	4,00
3	2,32	2,32
	Аналітичний метод згідно з [2]	Імітаційна модель
Загальний середній час перебування в магазині, хв	28,64	28,03

Незначна розбіжність результатів простежується лише для характеристик першої фази обслуговування. Це можна пояснити похибками обчислень аналітичним методом, які є досить громіздкими для чотириканальної СМО.

Приклад 5.16. *Процес виготовлення деталей на підприємстві потребує виконання трьох операцій на певних видах станків. Середній час, через який заготовки деталей надходять на виготовлення – 15 хв, відхилення від середнього часу ± 3 хв. Середній час, який витрачається на виконання кожної операції, і можливі відхилення від середнього часу відповідно дорівнюють: 16 ± 1 , 20 ± 3 і 8 ± 3 хв. Середній час, який витрачають для того, щоб передати деталь від одного станка до другого, і можливі відхилення від середнього часу відповідно дорівнюють 4 ± 1 хв і 5 ± 1 хв. Розподіл всіх заданих випадкових величин рівномірний. Визначити:*

- коефіцієнти використання всіх станків;
- середній час перебування деталі біля кожного станка;
- максимальну кількість деталей, які очікують виготовлення біля кожного станка;
- середню довжину черги, тобто середню кількість деталей, які очікують виготовлення біля кожного станка;
- середній час перебування деталей у чергах;
- загальну кількість деталей, які надійшли на обслуговування протягом зміни.

; Приклад 5.16.

CHSMOD EQU 480 ; час моделювання

GENERATE 15,3

; перша фаза обслуговування

QUEUE 1

SEIZE STAN1

DEPART 1

ADVANCE 16,1

RELEASE STAN1

ADVANCE 4,1

; друга фаза обслуговування

QUEUE 2

SEIZE STAN2

```

DEPART 2
ADVANCE 20,3
RELEASE STAN2
*****
ADVANCE 5,1
*****
; третя фаза обслуговування
*****
QUEUE 3
SEIZE STAN3
DEPART 3
ADVANCE 8,3
RELEASE STAN3
*****
TERMINATE
*****
GENERATE CHSMOD ; реалізувати час моделювання
*****
; обчислення середнього часу перебування деталі
; біля кожного станка
*****
SAVEVALUE PER1,(FT$STAN1+QT1)
SAVEVALUE PER2,(FT$STAN2+QT2)
SAVEVALUE PER3,(FT$STAN3+QT3)
*****
TERMINATE 1
START 1

```

За одиницю модельного часу в моделі прийнято 1 хв. Час моделювання дорівнює тривалості однієї зміни: 8 год=480 хв.

Інформацію про коефіцієнти використання станків можна знайти у стандартному звіті моделі серед характеристик одноканальних пристроїв STAN1, STAN2, STAN3. Статистика всіх черг також міститься в стандартному звіті як інформація про об'єкти QUEUE 1, QUEUE 2 і QUEUE 3. Загальну кількість деталей, які надійшли на обслуговування протягом зміни визначаємо серед характеристик об'єкта QUEUE 1 як загальну кількість входжень у першу чергу (ENTRY). Середній час перебування деталі біля кожного станка знайдено як суму середнього часу перебування в черзі (СЧА QT_j) та середнього часу обслуговування одного замовлення (СЧА FT\$STAN_j).

Результати моделювання зведено у табл. 5.9.

Таблиця 5.9

	Номер станка		
	1	2	3
Коефіцієнт використання станка	0,964	0,921	0,343
Середній час перебування деталі біля станка, хв	26,617	62,864	7,850
Найбільша довжина черги	2	6	1
Середня довжина черги	0,688	2,494	0,000
Середній час перебування деталі у черзі, хв	10,653	42,764	0,000

Загальна кількість деталей, які надійшли на обслуговування протягом зміни – 31.

6. МОДЕЛІ ЗАМКНЕНИХ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

Досі ми розглядали розімкнені СМО, для яких інтенсивність вхідного потоку не залежала від стану системи, а самі джерела замовлень перебували поза системою, тому їхні стани аналізу не підлягали. Тепер розглянемо **замкнені СМО**, в яких інтенсивність потоку замовлень залежить від стану системи, а джерела замовлень є внутрішніми елементами системи. Такі випадки можливі, коли система обслуговує обмежену кількість „клієнтів” (джерел замовлень), яка незначно відрізняється від кількості каналів обслуговування. Перебуваючи на обслуговуванні, „клієнт” перестає подавати замовлення, а після завершення обслуговування знову стає джерелом замовлень.

6.1. Замкнена система з одним каналом

Розглянемо аналітичну модель одноканальної замкненої СМО.

Нехай кожен з m однакових *технічних пристроїв* (ТП) може в деякі випадкові моменти часу потребувати обслуговування (відмовити, вийти з ладу тощо). Потік відмов кожного ТП має інтенсивність λ . Кожен ТП може обслуговувати один канал. Інтенсивність потоку обслуговувань – μ . Якщо в момент відмови ТП канал зайнятий, то цей ТП стає в чергу на обслуговування.

Стани системи зв'язуємо з кількістю ТП, які вийшли з ладу: x_1 – вийшов з ладу один ТП і він перебуває на обслуговуванні; x_{1+r} – вийшли з ладу $1+r$ ТП ($r = \overline{1, m-1}$), з них один перебуває на обслуговуванні і r очікують на обслуговування. Отже, така система має $m+1$ стан:

$$x_0 \underset{\mu}{\square} x_1 \underset{\mu}{\square} \cdots \underset{\mu}{\square} x_k \underset{\mu}{\square} \cdots x_{m-1} \underset{\mu}{\square} x_m.$$

$m\lambda$ $(m-1)\lambda$ $(m-k+1)\lambda$ $(m-k)\lambda$ λ

За допомогою графа можемо записати систему рівнянь для ймовірностей станів:

$$\begin{aligned} \frac{dp_0}{dt} &= -m\lambda p_0 + \mu p_1; \\ \frac{dp_1}{dt} &= -(\mu + (m-1)\lambda) p_1 + m\lambda p_0 + \mu p_2; \\ \frac{dp_{1+r}}{dt} &= -(\mu + (m-1-r)\lambda) p_{1+r} + (m-r)\lambda p_r + \\ &\quad + \mu p_{r+2}, \quad r = \overline{1, m-2}; \\ \frac{dp_m}{dt} &= -\mu p_m + \lambda p_{m-1}. \end{aligned} \tag{6.1}$$

Приклад 6.1. У гаражі 20 автомобілів і одне місце для проведення ремонту. В середньому кожен автомобіль протягом місяця тричі потребує ремонту, а ремонт одного автомобіля триває в середньому одну добу. Вважаючи, що потоки відмов і потік обслуговувань – найпростіші, визначити

середню кількість автомобілів, які перебувають на ремонті; середню кількість автомобілів, які чекають на ремонт; середню кількість автомобілів, які простоюють, і ймовірність простою окремого автомобіля. Побудувати імітаційну модель цієї замкненої СМО, а також модель GPSS World для числового інтегрування системи диференціальних рівнянь для ймовірностей станів. Порівняти результати, отримані за допомогою аналітичної моделі та моделей GPSS World.

Маємо одноканальну замкнену систему з параметрами:

$$m = 20; \quad \lambda = \frac{3}{30} = 0,1 \text{ доба}^{-1}; \quad \mu = 1 \text{ доба}^{-1}.$$

У випадку $m=20$ система рівнянь (6.1) набуває вигляду:

$$\begin{aligned} \frac{dp_0}{dt} &= -20\lambda p_0 + \mu p_1; \\ \frac{dp_{1+r}}{dt} &= -(\mu + (19-r)\lambda) p_{1+r} + (20-r)\lambda p_r + \\ &\quad + \mu p_{r+2}, \quad r = \overline{0,18}; \\ \frac{dp_{20}}{dt} &= -\mu p_{20} + \lambda p_{19}. \end{aligned} \tag{6.2}$$

Запишемо результати, отримані для цієї задачі аналітичним методом [4], які відповідають стаціонарному режиму роботи СМО:

$$\begin{aligned} p_0 &= 0,002, \quad \bar{s} = 1 - p_0 = 0,998; \quad \bar{l} = m - \mu(1 - p_0) / \lambda = 10,019; \\ \bar{r} &= \bar{l} - \bar{s} = 9,021; \quad \pi_{np} = \frac{\bar{l}}{m} = 0,501. \end{aligned} \tag{6.3}$$

Тут \bar{s} – середня кількість автомобілів, які перебувають на обслуговуванні; \bar{r} – середня кількість автомобілів, які очікують на обслуговування; \bar{l} – середня кількість автомобілів, які простоюють; π_{np} – ймовірність простою окремого автомобіля (коефіцієнт простою техніки).

Побудуємо моделі GPSS World.

; Приклад 6.1. Імітаційна модель замкненої СМО.

HAR STORAGE 1 ; одноканальна СМО

CHMOD EQU 2000000 ; час моделювання

GENERATE ,,20 ; створити 20 транзактів

ADV ADVANCE (Exponential(1,0,10)) ; потік відмов ТП

QUEUE 1

ENTER HAR

DEPART 1

ADVANCE (Exponential(1,0,1)) ; час обслуговування

LEAVE HAR

TRANSFER ,ADV ; стати джерелом відмов

GENERATE CHMOD ; реалізувати час моделювання

; обчислення характеристик СМО за результатами імітаційного
; моделювання

SAVEVALUE LSIM,(SA\$HAR+QA1) ; сер. к-ть ТП, які
; простоюють

SAVEVALUE PIIM,(X\$LSIM/20) ; імовірність простою

TERMINATE 1

START 1

; Приклад 6.1. Інтегрування системи диференціальних рівнянь
; для ймовірностей станів.

La EQU 1/10 ; інтенсивність потоку відмов

Mu EQU 1 ; інтенсивність потоку обслуговувань

CHMOD EQU 50 ; час моделювання

; початкові умови

p0_ EQU 1

p1_ EQU 0

p2_ EQU 0

p3_ EQU 0

p4_ EQU 0

p5_ EQU 0

p6_ EQU 0

p7_ EQU 0

p8_ EQU 0

p9_ EQU 0

p10_ EQU 0

p11_ EQU 0

p12_ EQU 0

p13_ EQU 0

p14_ EQU 0

p15_ EQU 0

p16_ EQU 0

p17_ EQU 0

p18_ EQU 0

p19_ EQU 0

p20_ EQU 0

p0_ INTEGRATE (-20#La#p0_+Mu#p1_)

p1_ INTEGRATE ((-Mu-19#La)#p1_+20#La#p0_+Mu#p2_)

p2_ INTEGRATE ((-Mu-18#La)#p2_+19#La#p1_+Mu#p3_)

простою автомобіля, як частку від ділення вмісту комірки, де зберігається значення середньої кількості автомобілів, які простоюють (X\$LSIM), на кількість автомобілів.

Для досягнення граничного стаціонарного режиму роботи системи розрахунки проведено для часу моделювання 2000000. Одиниця модельного часу – одна доба.

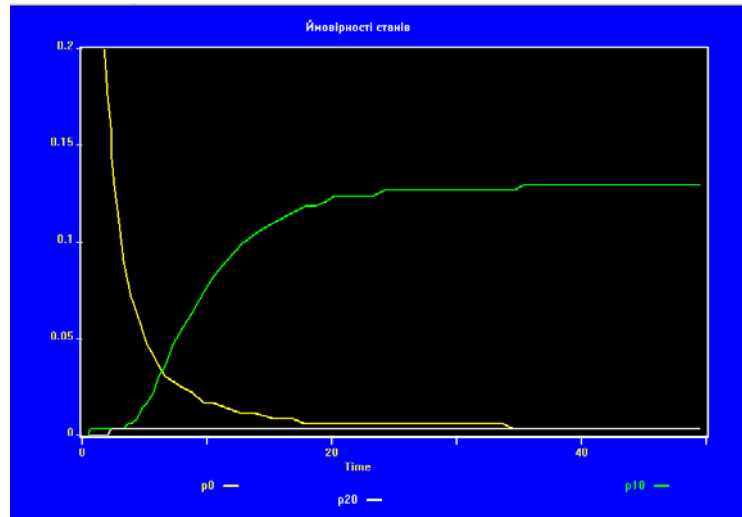


Рис. 6.1. Залежності ймовірностей p_0, p_{10} і p_{20} від часу, отримані числовим інтегруванням рівнянь аналітичної моделі

Числове інтегрування системи диференціальних рівнянь для ймовірностей станів (6.2) здійснено для значення часу моделювання 50, оскільки для такого значення часу стаціонарні значення ймовірностей станів вже досягаються (див. рис. 6.1). За результатами інтегрування характеристики замкненої СМО знайдено з використанням блоків SAVEVALUE за формулами (6.3).

Результати моделювання для порівняння зведено у табл. 6.1.

Таблиця 6.1

	\bar{s}	\bar{l}	\bar{r}	π_{np}
Аналітична модель	0,998	10,019	9,021	0,501
Числове інтегрування	0,998	10,020	9,022	0,501
Імітаційна модель	0,998	10,021	9,023	0,501

6.2. Багатоканальна замкнена система

Побудуємо імітаційну модель двоканальної замкненої СМО з рівномірним розподілом інтервалу відмов і часу обслуговування. Для такої СМО немає аналітичної моделі.

Приклад 6.2. *Розв'язати задачу з попереднього параграфа за допомогою імітаційної моделі, розглянувши випадки наявності в гаражі двох і трьох місць для ремонту. В середньому кожен автомобіль через кожні десять діб потребує ремонту, можливе відхилення від середнього значення ± 3 . Ремонт одного автомобіля триває в середньому одну добу, відхилення від середнього часу $\pm 0,5$ доби. Розподіл інтервалу відмов і часу обслуговування рівномірний.*

Беремо за одиницю модельного часу одну добу.

; Приклад 6.2. Імітаційна модель багатоканальної замкненої СМО.

HAR1 STORAGE 2 ; двоканальна СМО

HAR2 STORAGE 3 ; триканальна СМО

CHMOD EQU 100000 ; час моделювання

; випадок двох каналів

GENERATE ,,20 ; створити 20 транзактів

SPLIT 1,ADV2 ; створити копію транзакта

ADV1 ADVANCE 10,3 ; потік відмов ТП

QUEUE 1

ENTER HAR1

DEPART 1

ADVANCE 1,0.5 ; час обслуговування

LEAVE HAR1

TRANSFER ,ADV1 ; стати джерелом відмов

; випадок трьох каналів

ADV2 ADVANCE 10,3 ; потік відмов ТП

QUEUE 2

ENTER HAR2

DEPART 2

ADVANCE 1,0.5 ; час обслуговування

LEAVE HAR2

TRANSFER ,ADV2 ; стати джерелом відмов

GENERATE CHMOD ; реалізувати час моделювання

; обчислення характеристик двоканальної СМО

6.2. Багатоканальна замкнена система

```

*****
SAVEVALUE LSIM1,(SA$HAR1+QA1) ; середня кількість ТП, які
;                               простоюють
SAVEVALUE PIIM1,(X$LSIM1/20) ; імовірність простою
*****
; обчислення характеристик триканальної СМО
*****
SAVEVALUE LSIM2,(SA$HAR2+QA2) ; середня кількість ТП, які
;                               простоюють
SAVEVALUE PIIM2,(X$LSIM2/20) ; імовірність простою
*****
TERMINATE 1
START 1
    
```

Модель відрізняється від моделі з попереднього параграфу заданням двоканального і триканального пристроїв за допомогою блоків STORAGE, роботою блоків ADVANCE у режимі рівномірного розподілу і наявністю додаткового сегмента моделі, який імітує роботу триканальної СМО. 20 транзактів, згенерованих блоком GENERATE, одночасно потрапляють до блоків ADVANCE з мітками ADV1 і ADV2 завдяки наявності блока SPLIT.

Результати моделювання для двох значень часу моделювання зведено у табл. 6.2. Позначення у верхньому рядку таблиці такі самі як у табл. 6.1.

Таблиця 6.2

Час моделювання	\bar{s}	\bar{l}	\bar{r}	π_{np}
	Двоканальна система			
100000	1,726	2,739	1,013	0,137
1000000	1,726	2,738	1,012	0,137
	Триканальна система			
100000	1,799	1,995	0,196	0,100
1000000	1,800	1,998	0,198	0,100

Отже, ефективність системи підвищується зі збільшенням кількості каналів.

6.3. Багатофазна замкнена система

Розглянемо моделювання багатофазної замкненої системи на прикладі двофазної виробничої системи.

Приклад 6.3. На домобудівному комбінаті (ДБК) кран завантажує залізобетонні блоки на спеціальні вантажівки, а на будівельному об'єкті другий кран здійснює розвантаження блоків. Середній час завантаження машини краном на ДБК становить 15 хв, а можливі відхилення від середнього часу – 2 хв. Середній час, необхідний для транспортування вантажу на будмайданчик, становить 35 хв, можливе відхилення від цього часу – 2 хв. Середній час розвантаження вантажівки другим краном на будмайданчику становить 10 хв, можливе відхилення від нього – 1 хв. Середній час, необхідний для повернення машини на ДБК, становить 25 хв, а можливе відхилення від нього – 2 хв. Кількість машин, зайнятих транспортуванням блоків на будмайданчик – 7. Розподіли всіх випадкових величин рівномірні.

Визначити основні характеристики системи:

- 1) коефіцієнти використання першого і другого каналів обслуговування;
- 2) середній час перебування машини на обслуговуванні першим і другим краном;
- 3) максимальну довжину черг, тобто кількість машин, які очікують на завантаження та розвантаження;
- 4) середню довжину черг, тобто середню кількість машин, які очікують обслуговування (завантаження, розвантаження);
- 5) середній час перебування машин у чергах;
- б) загальну кількість обслужених машин протягом зміни.

Беремо за одиницю модельного часу 1 хв. Одна зміна триває 480 хв.

; Приклад 6.3. Імітаційна модель двофазної замкненої СМО.

; перша фаза обслуговування

```

GENERATE ,,7 ; створити сім транзактів
ZAV QUEUE 1
SEIZE KR1
DEPART 1
ADVANCE 15,2 ; час обслуговування
RELEASE KR1

```

; транспортування вантажу

```

ADVANCE 35,2 ; час транспортування

```

; друга фаза обслуговування

```

QUEUE 2
SEIZE KR2
DEPART 2
ADVANCE 10,1 ; час обслуговування
OBS RELEASE KR2
*****
; повернення в ДСК
*****
ADVANCE 25,2 ; час поїздки
TRANSFER ,ZAV ; стати у чергу на завантаження
*****
GENERATE 480 ; реалізувати час моделювання
TERMINATE 1
START 1

```

Ця модель відрізняється від моделей однофазних замкнених систем наявністю двох фаз обслуговування. Сегменти моделі, які імітують роботу цих фаз, розташовані послідовно. Випадкові інтервали часу транспортування вантажу та повернення вантажівки до ДБК імітуються за допомогою двох блоків ADVANCE. Блок TRANSFER, який повертає транзакт у першу чергу після повернення з другої, розташований після другого із зазначених блоків ADVANCE.

Всі характеристики системи, які необхідно визначити згідно з умовою задачі, виводяться у стандартному звіті моделі. Більшість з них міститься серед інформації про роботу об'єктів QUEUE 1, QUEUE 2, FACILITY KR1 та FACILITY KR2. Кількість обслужених машин визначаємо за кількістю входжень до блока RELEASE KR2.

Результати моделювання зведено у табл. 6.3, де характеристики системи позначено згідно з нумерацією в умові прикладу.

Таблиця 6.3

Характеристики системи	Перша фаза	Друга фаза
1	1,000	0,612
2, хв	15,000	10,127
3	6	1
4	1,775	0,000
5, хв	25,064	0,000

Результати показують, що другий кран завантажений лише на 61,2%, і черги до нього немає. Протягом зміни обслужено 29 машин.

ДОДАТКИ

Додаток 1

Основні оператори мови GPSS

Оператор	Операнди	Опис оператора
GENERATE	A,B,C,D,E,F,G,H	Згенерувати транзакт
QUEUE	A,B	Ввести транзакт у чергу
DEPART	A,B	Вивести транзакт з черги
SEIZE	A	Зайняти канал обслуговування
RELEASE	A	Звільнити канал обслуговування
ENTER	A,B	Зайняти весь або частину багатоканального пристрою
LEAVE	A,B	Звільнити весь або частину багатоканального пристрою
TERMINATE	A	Знищити транзакт
ALTER		Перевірити і модифікувати транзакти у групі
ASSEMBLE	A	Об'єднати сім'ю транзактів в один транзакт
ASSIGN	A,B	Модифікувати параметр транзакта
BUFFER		Зайняти місце в кінці списку поточних подій
<Ім'я>BVARIABLE	A	Визначити булеву змінну
CLEAR		Скинути до нуля всю статистику і вилучити транзакти
COUNT	X A,B,C,D,E	Визначити кількість об'єктів, які задовольняють певну умову, і записати в параметр транзакта
DISPLACE	A,B,C,D	Знайти транзакт і перемістити до нового блока
EQU	A,B	Присвоїти значення

Оператор	Операнди	Опис оператора
EXAMINE		Дослідити належність транзакта до групи
EXECUTE	A	Примусово опрацювати транзакт деяким блоком
FAVAIL	A	Відновити доступність ОКП
FUNAVAIL	A,B,C,D,E,F,G,H	Встановити недоступність ОКП
<Ім'я>FUNCTION	A,B	Визначити функцію
<Ім'я>FVARIABLE	A	Визначити змінну з плаваючою крапкою
GATE	X A,B	Змінити напрям руху залежно від стану об'єкта
GATHER	A	Зібрати задану кількість транзактів одної сім'ї
INDEX	A,B	Змінити параметр транзакта
INITIAL	A,B	Присвоїти або модифікувати значення комірки або матриці (елемента матриці)
LINK	A,B,C	Ввести у список користувача
LOGIC	X A	Змінити стан логічного ключа
LOOP	A,B	Організувати цикл зі зменшенням значення параметра транзакта
MARK	A	Записати значення абсолютного модельного часу в параметр транзакта
MATCH	A	Синхронізувати транзакти
MATRIX	A,B,C,D,E,F,G	Визначити матрицю
MSAVEVALUE	A,B,C,D	Модифікувати значення елементів матриць
PREEMPT	A,B,C,D,E	Захопити пристрій активним таранзактом з перериванням
PRIORITY	A,B	Присвоїти пріоритет активному транзактові

Оператор	Операнди	Опис оператора
QTABLE	A,B,C,D	Визначити таблицю для черги
RESET		Скинути до нуля всю статистику, крім генераторів випадкових чисел і таймера абсолютного часу
RETURN	A	Звільнити захоплений пристрій
RMULT	A,B,C,D,E,F,G	Задати початкові значення генераторів випадкових чисел
SAVAIL	A	Відновити доступність БКП
SAVEVALUE	A,B	Модифікувати значення комірки зберігання
SELECT	X A,B,C,D,E,F	Вибрати перший об'єкт у заданому діапазоні, який задовольняє певну умову
SPLIT	A,B,C	Створити необхідну кількість копій активного транзакта
<Ім'я>STORAGE	A	Задати місткість БКП
SUNAVAIL	A	Перевести БКП у недоступний стан
<Ім'я>TABLE	A,B,C,D	Визначити таблицю
TABULATE	A,B	Зібрати дані в таблицю заданого імені
TEST	X A,B,C	Скерувати транзакт залежно від результатів порівняння операндів
TRANSFER	A,B,C,D	Скерувати активний транзакт до вказаного блока
UNLINK	X A,B,C,D,E	Вивести зі списку користувача
<Ім'я>VARIABLE	A	Визначити дійсну змінну

Системні числові атрибути GPSS

Код	Значення	Примітка
<i>Системні числові атрибути</i>		
RNj	Випадкове число. Цілочисельне значення.	Датчик генерує послідовність рівномірно розподілених випадкових чисел від 0 до 999.
C1	Відносний модельний час. Дійсне значення.	Модельний час від моменту останньої команди RESET. Автоматично змінюється системою.
AC1	Абсолютний модельний час. Дійсне значення.	Модельний час від моменту останньої команди CLEAR. Автоматично змінюється системою.
TG1	Число, яке дорівнює поточному значенню лічильника завершень.	Початкове значення TG1 задає оператор START. Рівність нулю значення TG1 свідчить про завершення моделювання. Транзакти, які ввійшли до блоків TERMINATE з ненульовим додатним операндом A, зменшують це значення на число, яке дорівнює значенню операнда A.
Z1	Поточна вільна системна пам'ять. Цілочисельне значення.	
<i>Системні числові атрибути транзактів</i>		
Pj	Значення j-го параметра активного транзакта. Цілочисельне, дійсне або рядкове значення.	Для непрямої адресації використовуються: СЧА*j; СЧА*ім'я параметра; СЧА*\$ім'я параметра.

Код	Значення	Примітка
XN1	Номер активного транзакта. Цілочисельне значення.	
PR	Пріоритет активного транзакта. Цілочисельне значення.	Може змінюватися блоком PRIORITY. За замовчуванням пріоритет дорівнює 0.
MBj	Відповідність у блоці MATCH. Цілочисельне значення.	1, якщо у спряженому блоці MATCH перебуває блок тої самої сім'ї, що й активний транзакт; інакше – 0.
MPj	Час проходження транзактом деякої частини моделі. Дійсне значення.	Обчислюють як різницю поточного абсолютного модельного часу і значення j-го параметра активного транзакта. Різниця – транзитний час – зберігається у j-му параметрі.
M1	Час перебування в моделі активного транзакта. Дійсне значення.	Обчислюють як різницю абсолютного модельного часу і відмітки часу активного транзакта, в якій зберігається або час надходження транзакта до моделі, або час останнього проходження транзактом блока MARK без операнда A.
<i>Системні числові атрибути блоків</i>		
Nj	Загальна кількість входжень транзактів до j-го блока. Цілочисельне значення.	Збільшується з кожним входженням транзакта до j-го блока.
Wj	Поточна кількість транзактів, які перебувають у j-му блоці. Цілочисельне значення.	Збільшується з кожним входженням транзакта до j-го блока і зменшується з кожним виходом транзакта з j-го блока.

Код	Значення	Примітка
<i>Системні числові атрибути БКП</i>		
Sj	Поточний вміст пристрою j. Цілочисельне значення.	Змінюється блоками ENTER і LEAVE.
Rj	Кількість вільних одиниць пристрою j. Цілочисельне значення.	Змінюється блоками ENTER і LEAVE.
SRj	Коефіцієнт використання пристрою j. Дійсне значення.	Виражається у тисячних частках, але значення СЧА – ціле, тобто якщо коефіцієнт дорівнює 0,65, то SRj дорівнює 650. Може бути нецілочисельним.
SAj	Середня кількість зайнятих каналів пристрою j. Дійсне значення.	Середня за одиницю часу кількість зайнятих каналів пристрою.
SMj	Максимальна кількість зайнятих каналів пристрою j. Цілочисельне значення.	Максимальна кількість каналів пристрою, які були зайняті одночасно у процесі моделювання.
SCj	Лічильник використання пристрою j. Цілочисельне значення.	Загальна кількість входжень до БКП. Збільшується з кожним входженням транзакта до блока ENTER.
STj	Середній час використання одного каналу пристрою j.	Система визначає автоматично.
SEj	Зайнятість пристрою j.	1 – не зайнятий; 0 – зайнятий.
SFj	Заповненість пристрою j.	1 – заповнений; 0 – не заповнений.
SVj	Доступність пристрою j.	1 – доступний; 0 – недоступний.
<i>Системні числові атрибути ОКП</i>		
Fj	Зайнятість пристрою j. Цілочисельне значення.	1 – зайнятий; 0 – не зайнятий. Може бути змінений блоками SEIZE, RELEASE, PREEMPT і RETURN.

Код	Значення	Примітка
FIj	Перерваність пристрою j. Цілочисельне значення.	1 – перерваний; 0 – не перерваний. Може бути змінений блоками PREEMPT і RETURN.
FVj	Доступність пристрою j. Цілочисельне значення.	1 – доступний; 0 – недоступний. Може бути змінений блоками FAVAIL і FUNAVAIL.
FRj	Коефіцієнт використання пристрою j.	Виражається у тисячних частках, але значення СЧА – ціле, тобто якщо коефіцієнт дорівнює 0,88, то FRj дорівнює 880.
FCj	Кількість транзактів, які займали пристрій j.	Збільшується з входженням транзакта до блока SEIZE або PREEMPT.
FTj	Середній час використання пристрою j одним транзактом.	Система визначає автоматично.
<i>Системні числові атрибути черг</i>		
Qj	Поточна довжина черги j. Цілочисельне значення.	Збільшується із входженням транзакта до блока QUEUE і зменшується із входженням до блока DEPART.
QAj	Середня довжина черги j.	Середня за одиницю часу кількість транзактів у черзі j.
QMj	Максимальна довжина черги j. Цілочисельне значення.	Максимальна кількість транзактів, які одночасно перебували в черзі j.
QCj	Загальна кількість входжень у чергу j. Цілочисельне значення.	Автоматично збільшується системою із входженням транзакта до черги j.
QZj	Кількість входжень з нульовим часом перебування в черзі j.	Кількість входжень транзактів, час перебування яких у черзі дорівнює нулю.
QTj	Середній час перебування транзакта в черзі j.	Обчислюють із урахуванням нульових входжень.

Код	Значення	Примітка
QXj	Середній час перебування транзакта в черзі j.	Обчислюється без урахування нульових входжень.
<i>Системні числові атрибути комірок і матриць комірок</i>		
Xj	Вміст комірки j. Цілочисельне, дійсне або рядкове значення.	Змінюється блоком SAVEVALUE.
MXj (a,b)	Вміст комірки матриці j. Цілочисельне, дійсне або рядкове значення.	Значення комірки в рядку a і в стовпці b. Змінюється блоком SAVEVALUE. Для матриць, які мають більше двох вимірів, всі інші індекси дорівнюють 1. На відміну від СЧА класу MX, PLUS-вирази можуть звертатися до будь-якого елемента багатовимірних матриць.
<i>Системні числові атрибути обчислювальних об'єктів</i>		
FNj	Результат обчислення значення функції j. Дійсне значення.	
Vj	Результат обчислення цілочисельної змінної j або змінної j з плаваючою крапкою. Дійсне значення.	
BVj	Результат обчислення булевої змінної j. Дійсне значення.	Набуває значень: 1 (true); 0 (false).
<i>Системні числові атрибути груп, списків та ключів</i>		
GNj	Поточна кількість членів у j-й числовій групі.	
GTj	Поточна кількість членів у j-й групі транзактів.	

Код	Значення	Примітка
CHj	Поточна кількість членів у j-му списку користувача.	Список користувача створює сам програміст на відміну від інших списків моделі. Використовують блоки LINK та UNLINK. Значення всіх СЧА визначаються аналогічно значенням СЧА черг.
CAj	Середня кількість транзактів у j-му списку користувача.	
CMj	Максимальна кількість транзактів у j-му списку користувача.	
CSj	Загальна кількість транзактів у j-му списку користувача.	
CTj	Середній час перебування транзакта у j-му списку користувача.	
LSj	Стан j-го логічного ключа.	Набуває значень: 1 – установлений; 0 – не установлений.
<i>Системні числові атрибути таблиць</i>		
TVj	Середнє значення неусередненого аргументу таблиці j.	Для занесення в таблицю використовують блок TABULATE.
TCj	Кількість введень у таблицю j.	Цілочисельне значення.
TDj	Середнє квадратичне відхилення аргументу таблиці j.	Дійсне значення.

СПИСОК ЛІТЕРАТУРИ

1. *Боев В. Д.* Моделирование систем. Инструментальные средства GPSS World. – Санкт-Петербург, 2004.
2. *Вентцель Е. С., Овчаров Л. А.* Прикладные задачи теории вероятностей. – М., 1983.
3. *Гнеденко Б. В.* Курс теории вероятностей. – М., 1988.
4. *Жерновий Ю. В.* Марковські моделі масового обслуговування. – Львів, 2004.
5. *Ивченко Г. И., Каштанов В. А., Коваленко И. Н.* Теория массового обслуживания. – М., 1982.
6. *Кудрявцев Е. М.* GPSS World. Основы имитационного моделирования различных систем. – М., 2004.
7. *Томашевский В. Н., Жданова Е. Г.* Имитационное моделирование в среде GPSS. – М., 2003.
8. *Cooper R. B.* Introduction to queueing theory. – New York, 1981.

ПРЕДМЕТНИЙ ПОКАЖЧИК

- Адресація
 непряма, 31
 пряма, 31
- Бібліотечні генератори випадкових чисел, 34
- Бібліотечні математичні функції, 33
- Блок
 ADVANCE, 64
 ASSIGN, 59
 DEPART, 86
 DISPLACE, 85
 ENTER, 67
 GATE, 70
 GENERATE, 48
 INDEX, 60
 LEAVE, 67
 LINK, 101
 LOGIC, 100
 LOOP, 83
 MARK, 60
 MSAVEVALUE, 43
 PLUS, 59
 PRIORITY, 107
 QUEUE, 86
 RELEASE, 62
 SAVAIL, 94
 SAVEVALUE, 40
 SEIZE, 62
 SPLIT, 194
 SUNAVAIL, 94
 TABULATE, 89
 TERMINATE, 56
 TEST, 72
 TRANSFER, 74
 UNLINK, 102
- Блок-діаграма моделі, 27
- Блоки GPSS World, 22
- Вилучення транзактів, 56
- Вирази, 39
- Вікна GPSS World, 123
- Вікно
 блока GENERATE, 111
 головне GPSS World, 108
 діалогове команди START, 113
 діалогове Command ► SHOW..., 127
 діалогове Edit Plot, 130
 діалогове Open Table, 132
 журналу налаштувань, 125
 текстового редактора, 110
 Blocks, 126
 Command ► Custom..., 126
 Edit Expression, 129
 Expression, 129
 Journal, 112, 113, 114
 Notices, 108
 Plot, 131
 Queues, 127
 Stop Conditions, 133
 Storages, 128
 Table, 133
- Генерування випадкових чисел, 34, 51
- Годинник (таймер), 14, 21, 28
- Головне вікно GPSS World, 108
- Дані
 дійсні, 30
 цілочисельні, 30
- Дисципліна обслуговування, 13
 LIFO, 13
 FIFO, 13
 RANDOM, 13
- Дієслово, 28
- Дія, 14
- Затримка, 14
- Звіт стандартний, 138
- Змінна, 23
 арифметична, 23, 44

- Змінна
булева, 24, 46
з плаваючою крапкою, 45
користувача, 38
- Імовірність обслуговування, 146,
152, 161
- Імовірності станів, 145, 151, 213
- Ім'я об'єкта, 29
- Інтегрування числове, 144
- Інтервал надходження, 48
- Категорії об'єктів, 21
- Команда
BARIABLE, 46
CONDUCT, 117
CONTINUE, 117
CLEAR, 117
EQU, 42
EXIT, 118
FVARIABLE, 44
FUNCTION, 52
HALT, 119
INITIAL, 39
INCLUDE, 119
INTEGRATE, 144
MATRIX, 41
QTABLE, 89
REPORT, 119
RESET, 120
RMULT, 51
SHOW, 120
START, 57, 58
STEP, 121
STOP, 121
STORAGE, 67
TABLE, 88
- Коментар, 28
- Комірка зберігання, 39
- Константа, 30
- Ключ логічний, 23, 100
- Класифікація СМО, 11
- Лічильник завершень, 56
- Логічний оператор, 46
- Логічний ключ, 23, 100
- Меню блоків GPSS, 110
- Мітка, 28
- Моделювання
аналітичне, 7
багатофазних систем, 206, 220
виходу з ладу БКП, 96
дискретно-подієве, 14
замкнених систем, 213, 218, 220
збору статистики з кожного каналу, 186
імітаційне, 7, 14
недоступності пристрою, 94
обмежень на час очікування, 170
обслуговування неоднорідних замовлень, 176
обмежень на коефіцієнти використання каналів, 188
пріоритетного обслуговування, 176
сумарного потоку, 222
розрідженого потоку, 228
- Модель, 14
- Модельний час, 14
абсолютний, 28
відносний, 28
- Модифікатор-інтервал, 49
- Модифікатор-функція, 49
- Надходження транзактів, 48
- Найпростіший потік, 162
- Недоступність пристрою, 94
- Номер стрічки, 28
- Об'єкт, 14
GPSS, 21
- Операнд, 48
- Оператор, 21
арифметичний, 32
інтерактивний, 115

- Оператор
 умовний, 32
 логічний, 32, 46
 GPSS, 28
- Очікування
 безумовне, 15
 умовне, 15
- Перемикач, 23, 100
- Повідомлення про зупинку, 126
- Повідомлення про подію, 14
- Подія, 14, 27
 первинна, 15
- Поле оператора, 28
- Потік
 відмов, 213
 замовлень, 10
 найпростіший, 162
 нестационарний, 10
 ординарний, 10
 стаціонарний, 10
 сумарний, 222
 розріджений, 228
- Пристрій
 багатоканальний (БКП), 23
 одноканальний (ОКП), 22, 23, 48
- Програма-планувальник, 21
- Реєстратор черги, 86
- Режим
 безумовної передачі, 74
 одночасний, 82
 параметричний, 81
 підпрограмний, 82
 статистичної передачі, 75
 функціональний, 81
 ALL, 76
 BOTH, 77
 PICK, 79
- Середня кількість зайнятих каналів,
 146, 152
- Система, 14
- Система кодування СМО, 12
- Система масового обслуговування,
 10
 багатофазна, 12, 206
 без обмежень на довжину черги,
 157
 замкнена, 12, 213
 з відмовами, 71, 96, 146
 з відмовами і повторними
 спробами, 76, 83
 з відмовами класична, 144
 з обмеженням на довжину черги,
 151
 з очікуванням, 65, 68, 87, 90, 104,
 151, 170
 з пріоритетним обслуговуванням,
 105
 розімкнена, 12
 G/G/1/0, 161
- Система GPSS World, 8
- Системний числовий атрибут (СЧА),
 24, 30
- Список, 14
 відкладених переривань, 26
 затримки, 26
 користувача, 26, 101
 майбутніх подій, 14, 25
 переривань, 26
 повторних спроб, 25, 26
 подій, 14
 поточних подій, 25
 синхронізованих транзактів, 26
- Стан системи, 14
- Стандартний звіт, 138
- Таблиця, 24
 статистична, 88
- Таймер, 14, 21, 28
- Типи об'єктів, 21
- Транзакт, 21

Функція, 24, 33, 52
 дискретна атрибутивна (типу E),
 55
 дискретна числова, (типу D), 53
 неперервна числова (типу C), 54
 спискова атрибутивна (типу M),
 56
 спискова числова (типу L), 56

Черга, 24, 86

Числове інтегрування, 144