

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Литвиненко О.Є.
« _____ » _____ 2022 р.

**КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ
“МАГІСТР”**

Тема: «Імітаційна модель системи обслуговування з шкідливими запитами»

Виконавець: _____ Портянко О.О.

Керівник: _____ Коба О.В.

Консультанти з окремих розділів пояснювальної записки:

_____ Коба О.В.

Нормоконтролер: _____ Коба О.В.

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

(шифр, найменування)

Освітньо-професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

Литвиненко О.Є.

« » 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи (проєкту)

Портянко Олени Олександрівни

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи (проєкту): «Імітаційна модель системи обслуговування з шкідливими запитами»

затверджена наказом ректора від «16» 09 2022 р. № 1530/ст

2. Термін виконання роботи (проєкту): з 05.09.2022 по 30.11.2022

3. Вихідні дані до роботи (проєкту): система масового обслуговування з шкідливими запитами, пакет прикладних програм Matlab, мова програмування C++.

4. Зміст пояснювальної записки: _____

1) шкідливі запити, сфери їх впливу та різноманіття їх дій;

2) аналітичне та імітаційне моделювання систем;

3) аналітична модель системи масового обслуговування з шкідливими запитами;

4) імітаційна модель системи масового обслуговування з шкідливими запитами.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: _____

1) графи станів СМО;

2) розв'язання систем рівнянь Чепмена-Колмогорова;

3) діаграма класів розробленої імітаційної моделі;

4) алгоритм роботи імітаційної моделі;

5) результати роботи імітаційної моделі.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Ознайомитися з постановкою задачі кваліфікаційної роботи	05.09.2022-06.09.2022	
2	Опрацювати спеціальну літературу і технічну документацію	07.09.2022-09.09.2022	
3	Підготувати текст першого розділу пояснювальної записки	12.09.2022-22.09.2022	
4	Підготувати текст другого розділу пояснювальної записки	23.09.2022-28.09.2022	
5	Побудувати аналітичну модель систем	29.09.2022-06.10.2022	
6	Знайти аналітичні вирази показників ефективності систем за допомогою пакету Matlab	07.10.2022-10.10.2022	
7	Підготувати текст третього розділу пояснювальної записки	11.10.2022-14.10.2022	
8	Розробити алгоритм роботи системи	17.10.2022-19.10.2022	
9	Підготувати текст третього розділу пояснювальної записки	20.10.2022-25.10.2022	
10	Написати і налагодити програму	26.10.2022-10.11.2022	
11	Підготувати текст четвертого розділу пояснювальної записки	11.11.2022-13.11.2022	
12	Завершити оформлення пояснювальної записки	14.11.2022	
13	Пройти нормоконтроль, підготувати доповідь та презентацію до захисту	15.11.2022-21.11.2022	

7. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
1. Шкідливі запити	д.ф.-м.н., професор		

	Коба О.В.		
2. Аналітичне та імітаційне моделювання систем	д.ф.-м.н., професор Коба О.В.		
3. Аналітична модель СМО з шкідливими запитами	д.ф.-м.н., професор Коба О.В.		
4. Імітаційна модель СМО з шкідливими запитами	д.ф.-м.н., професор Коба О.В.		

8. Дата видачі завдання: « 05 » 09 2022 р.

Керівник кваліфікаційної роботи (проекту) _____ Коба О.В.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Портянко О.О.
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Імітаційна модель системи обслуговування з шкідливими запитами»: 95 с., 23 рис., 4 табл., 23 літературних джерела, 1 додаток.

СИСТЕМА ОБСЛУГОВУВАННЯ, ІМІТАЦІЙНА МОДЕЛЬ СИСТЕМИ ОБСЛУГОВУВАННЯ, АНАЛІТИЧНА МОДЕЛЬ СИСТЕМИ ОБСЛУГОВУВАННЯ, ШКІДЛИВІ ЗАПИТИ, СИСТЕМА ОБСЛУГОВУВАННЯ З ШКІДЛИВИМИ ЗАПИТАМИ, МОДЕЛЮВАННЯ СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ, СИСТЕМА З ВІДМОВАМИ, КАНАЛИ ОБСЛУГОВУВАННЯ, ОБСЛУГОВУВАННЯ ЗАПИТІВ.

Об'єкт дослідження: система масового обслуговування з шкідливими запитами.

Предмет дослідження: імітаційна модель системи масового обслуговування з шкідливими запитами.

Мета роботи: розробка імітаційної моделі системи масового обслуговування з шкідливими запитами.

Методи дослідження: аналітичний та чисельний методи, аналітичне моделювання систем масового обслуговування, імітаційне моделювання систем масового обслуговування, алгоритмізація, програмування, тестування.

Результати кваліфікаційної роботи рекомендується використовувати під час проектування та функціонування систем масового обслуговування з можливим підпотокком шкідливих запитів для ефективної експлуатації систем.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1 ШКІДЛИВІ ЗАПИТИ	13
1.1. Поняття шкідливих запитів.....	13
1.2. Різновиди шкідливих запитів.....	14
1.3. Сфери впливу шкідливих запитів.....	16
1.4. Різноманіття дій шкідливих запитів.....	17
1.5. Висновки до розділу	18
РОЗДІЛ 2 АНАЛІТИЧНЕ ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ ..	20
2.1. Поняття аналітичної моделі	25
2.2. Поняття імітаційної моделі	27
2.3. Висновки до розділу	35
РОЗДІЛ 3 АНАЛІТИЧНА МОДЕЛЬ СМО З ШКІДЛИВИМИ ЗАПИТАМИ	37
3.1. Одноканальна СМО	38
3.2. Двоканальна СМО.....	47
3.3. Висновки до розділу	70
РОЗДІЛ 4 ІМІТАЦІЙНА МОДЕЛЬ СМО З ШКІДЛИВИМИ ЗАПИТАМИ ..	72
4.1. Опис середовища програмування	72
4.2. Загальний принцип роботи програми	73
4.3. Опис програмної частини.....	74
4.4. Порівняння результатів аналітичної та імітаційної моделей	85
4.5. Висновки до розділу	92
ВИСНОВКИ.....	93
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	95
ДОДАТОК А.....	98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

СМО	- система масового обслуговування
ПЗ	- програмне забезпечення
ООП	- об'єктно-орієнтоване програмування
ЕОМ	- електронно-обчислювальна машина
ВВ	- випадкова величина

ВСТУП

У повсякденному житті люди часто мають справу з процесом обслуговування або ж із системами, які виконують цю роль. Для дослідження процесів, які виконуються в таких системах, використовують математичні об'єкти, які називаються системами масового обслуговування. Такі системи дуже популярні та часто вживані вже протягом багатьох років. Їх використовують в багатьох галузях, наприклад, сфери обслуговування, магазини, банківські і державні установи, телефонні служби, аеропорти, вокзали, підприємства та інше.

Кожна система з різноманіття систем має свій індивідуальний принцип обслуговування, який полягає в поведінці запиту в самій системі та поза нею, але глобально кожна з них схожа за складовими та суттю роботи, яку вони загалом всі виконують. Такі системи мають один або більше обслуговуючих каналів, які надають послуги запитам, які надходять до цієї системи. В ролі таких каналів можуть слугувати люди, які надають різні послуги, лінії зв'язку, різноманітні пристрої тощо. Діяльність таких систем обслуговування полягає в тому, щоб задовольнити потік з запитів, які надходять до системи масового обслуговування в випадкові проміжки часу. Обслуговування каналом кожного запиту займає деякий час під час якого він є зайнятим, а після оброблення канал звільняється і готовий до опрацювання наступного запиту, якщо такий є. Але потрібно також враховувати той факт, що до системи може надійти і шкідливий запит, як це й доволі часто трапляється у нашому повсякденному житті, який може вразити систему або ж навіть вивести її з ладу. Такий сценарій не є бажаним, але до нього слід бути готовим, адже це не нове явище, при такому ході подій канал, який вразить шкідливий запит, буде недоступний деякий час, він буде перебувати в режимі ремонту доти, доки він не буде поладжений і не буде готовий до коректного функціонування. При цьому система буде малоефективна, адже

якщо цей канал в системі один, то запити можуть накопичуватися в черзі тривалий час і не оброблятися вчасно або ж взагалі втрачатись, якщо ж канал не один, то така система теж не буде подавати багато надій, адже тим каналам необхідно буде справлятися не лише з своїми запитами, а й з тими, що могли б надходити на той канал обслуговування, що знаходиться на ремонті або ж якщо запитів буде мало, то багатьом каналам доведеться просто простоювати, що теж не дуже добре, але на відміну від попереднього випадку цей випадок більш вдалий, оскільки заявки не будуть втрачені. Схожих випадків є багато і зустрічаються вони нам дуже часто, тому доречно буде вивчити та дослідити режими функціонування систем обслуговування, розглянути типи шкідливих запитів, які можуть надійти до системи, різноманіття таких запитів та дії, які вони виконують.

Одною з задач роботи було проведення обрахунків з обчислення ймовірностей потрапляння шкідливого запиту до системи обслуговування та виведення її з ладу, такі задачі відносяться до розділу теорії ймовірності - теорії масового обслуговування. Теорія масового обслуговування вивчає ефективність роботи каналів систем обслуговування, досліджує явища, які виникають у процесі опрацювання запиту та функціонування самої системи масового обслуговування, продуктивність та ефективність системи в цілому.

Основною задачею роботи було зімітувати роботу системи масового обслуговування, на вхід якої надходять запити, серед яких можуть бути й шкідливі та розрахувати параметри ефективності такої системи, де значення будуть максимально точними до реальних.

Актуальність теми: атакування систем шкідливими запитами виникають у різноманітних сферах діяльності, захистити систему від таких запитів на жаль не вдасться, але за допомогою розробленої імітаційної моделі програмно можна буде відслідкувати роботу, поведінку та зміну станів системи, не завдаючи шкоди реальній системі, завдяки розробленій імітаційній моделі можна буде отримати показники ефективності функціонування системи масового обслуговування з шкідливими запитами

на етапі проектування та експлуатації системи з будь-якими неперервними розподілами часових інтервалів.

Мета виконання кваліфікаційної роботи: розробити імітаційну модель системи масового обслуговування з шкідливими запитами.

Завдання кваліфікаційної роботи: ознайомитися зі шкідливими запитами, з діями, які вони можуть завдавати, ознайомитися з різновидами шкідливих запитів, зі сферами їх впливу, з різноманіттям дій шкідливих запитів, розглянути СМО з одним каналом та двома каналами обслуговування з ймовірністю надходження шкідливого запиту в систему, обрахувати програмно та аналітично показники ефективності системи, здійснити програмну реалізацію СМО з шкідливими запитами.

В ході розгляду аналітичної моделі системи представлено дві СМО з одним та двома каналами обслуговування, на вхід яких надходить потік із запитами. Кожну з систем було розглянуто окремо, визначено всі можливі стани системи, для кожної з них побудовано граф станів з інтенсивностями переходів, обчислено системи лінійних алгебраїчних рівнянь шляхом власних розрахунків та за допомогою пакету Matlab, який полегшує обчислення за рахунок розв'язання задач будь-яких типів складності, витрачаючи при цьому мінімум часу.

Для обчислення рівнянь, які приведені в пояснювальній записці, було використано Matlab 2019. Для того, щоб розв'язати систему лінійних алгебраїчних рівнянь було використано вбудовані функції Matlab – `dsolve` та `solve` відповідно. Але перед тим необхідно попередньо перевірити наявність встановленого пакету, який має назву Symbolic Math Toolbox. Без його наявності скористатися цими функціями, на жаль, не вдасться. Їх можна буде встановити відразу ж при встановленні програмного пакету Matlab або ж додати після скачування програми. Symbolic Math Toolbox – це пакет, який містить набір різноманітних функцій для розв'язання, побудови та оброблення математичних рівнянь, які використовуються в таких математичних галузях, як алгебраїчні та диференціальні рівняння, лінійна

алгебра, спрощення рівнянь, виконання складних обчислень та інші взаємодії з рівняннями.

Основною ціллю роботи є симуляція роботи системи масового обслуговування на вхід якої може надходити шкідливий запит, як в реальній системі. Параметри СМО, такі як час обслуговування запиту або ж розмір черги близькі до реальних даних.

Об'єкт дослідження: система масового обслуговування з шкідливими запитами.

Предмет дослідження: імітаційна модель системи масового обслуговування з шкідливими запитами.

Методи дослідження: аналітичний та чисельний методи, аналітичне моделювання систем масового обслуговування, імітаційне моделювання систем масового обслуговування, алгоритмізація, програмування, тестування.

Наукова новизна отриманих результатів: розроблено імітаційну модель системи масового обслуговування з шкідливими запитами, яка може бути застосована для отримання показників ефективності функціонування системи масового обслуговування з шкідливими запитами та етапі проектування та експлуатації системи з будь-якими неперервними розподілами часових інтервалів.

Практичне значення отриманих результатів: при заданні початкових параметрів системи, програма моделює процес роботи СМО, всі процеси в ній виконуються в режимі реального часу з усіма можливими частинами, згідно з реальною СМО. Всі отримані результати роботи цієї моделі максимально наближені до еталонних даних, тому таку програму можна вважати ефективною та коректно виконуючою. Всі дані про канали обслуговування, стани системи, час перебування в станах та про систему загалом виводяться в файли, які можна відкривати та детально опрацьовувати для подальшої аналітики та інших необхідних видів діяльності.

Прогнозовані припущення щодо подальшого розвитку роботи. Дослідження у подальшому систем масового обслуговування з шкідливими запитами, в результаті яких шкідливі запити розповсюджуються на інші канали чи обладнання.

РОЗДІЛ 1

ШКІДЛИВІ ЗАПИТИ

Система масового обслуговування (СМО) - це система, яка складається з одного або більше каналів, які обслуговують запити, що надходять до системи.

Основними елементами СМО є:

- вхідний потік запитів;
- канали обслуговування;
- черга запитів;
- вихідний потік запитів.

Діяльність СМО полягає в обслуговуванні запитів, які один за одним надходять до системи, утворюючи при цьому вхідний потік, та які будуть обслуговуватися наявними каналами в системі, якщо вони будуть вільні, в іншому ж випадку такі запити можуть утворювати черги, які згодом також будуть оброблені та виведені з системи [1].

Вхідний потік СМО може складатися не лише з реальних запитів, але й з шкідливих.

1.1. Поняття шкідливих запитів

Шкідливі запити – це запити, які надходять в СМО та які містять в собі будь-який тип шкідливого програмного забезпечення (ПЗ). Такі запити можуть вивести з ладу канали обслуговування або й всю систему.

Шкідливе ПЗ – це програми, які розроблені для того, щоб причиняти шкоду системі, використовувати її в протизаконних цілях без відомості користувача, отримувати доступ до інформації системи або ж викрадати її.

Шкідливе ПЗ може:

- перехвачувати керування програмами;

- перенаправляти запити;
- відслідковувати дії в системі;
- викрадати особисті дані;
- використовувати систему для атак на інші системи.

Шкідливе ПЗ також може погіршувати роботу системи, впливати на процеси, які в ній відбуваються, погіршувати швидкість роботи, робити її нестабільною або ж аварійно завершувати роботу системи.

В подальшому ході виконання кваліфікаційної роботи буде розглянуто саме такі шкідливі запити, які, потрапляючи в систему, будуть виводити канал обслуговування з ладу.

1.2. Різновиди шкідливих запитів

Шкідливі запити можна поділити на дві групи, в першу входять ті, які не можуть існувати без деяких утиліт, системних програм або ж програмних додатків, в другу – ті, які можуть існувати незалежно від якихось програм, вони можуть плануватися та запускатися операційною системою.

До першої групи входять запити з: люком, логічною бомбою або троянським конем, до другої групи – запити з черв'яком, зомбі, програмою-викрадачем, поліморфік-генератори.

Люк – це прихований вхід до деякої програми системи, за допомогою люка можна отримати доступ до цієї програми. При наявності люка програма може викликатися нестандартними способами, що може негативно вплинути на саму систему, на її поведінку та захист самої системи.

Логічна бомба – це спеціальний шматок коду, який розміщений в деякій легальній програмі, який може «вибухнути» при якихось спеціальних умовах,

наприклад, наявність або відсутність якогось файлу або файлів, вхід в систему під деяким користувачем або ж навіть якась конкретна дата теж може вплинути на цей процес.

Троянський кінь – програма, яка на перший погляд здається корисною та виконує додаткові дії в додаток до основних, але вона містить прихований код, який може виконувати шкідливі дії в системі в разі його спрацювання. До цього типу можуть належати деякі люки, логічні бомби, утиліти прихованого адміністрування, програми-жарти, тобто вони можуть існувати у вигляді троянського коня або ж як самостійна програма.

Черв'як – програма, яка передається по мережі та не залишає своїх копій на магнітних носіях (магнітних стрічках, дисках, барабанах, картах). Ця програма розповсюджується з одної системи на іншу за допомогою мережевого з'єднання та використовує механізм підтримки мережі, за допомогою якого ця програма-черв'як визначає який з вузлів заражений і згодом «переповзає» до нього та активізується або ж чекає якоїсь сприятливої умови.

Зомбі – програма, яка приховано розповсюджується на під'єднані до мережі системи і від них здійснює різноманітні атаки, таким чином ускладнює спосіб виявлення справжньої шкідливої програми.

Програма-викрадач – спеціальна програма, створена для того, щоб викрадати дані (особисту інформацію, паролі тощо). Вона має вплив на набір даних для інших програм або ж для входу в систему, під час першої спроби ввійти в систему чи ввести якусь особисту інформацію ці дані викрадаються і спроба буде невдалою, хоча дані могли бути правильними, таким чином ця програма їх перехоплює, лише друга спроба буде вдалою за умови введення коректних даних.

Поліморфік-генератор – програма, яка шифрує тіло шкідливого запита і створює йому відповідний розшифровувач. Він не розмножується в системі, але за рахунок шифрування його дуже важко виявити. При надходженні запита з даною програмою виконується головний модуль генератора, який викликає функцію розшифрування і таким чином програма стає активною та завдає шкоди системі.

До того ж шкідливі запити можуть поділятися на такі, що розмножуються та не розмножуються. Шкідливі запити, які не здатні розмножуватися, активізуються в момент дії головної програми від якої вони залежні і виконують свою роботу не створюючи при цьому своїх копій, шкідливі ж запити, які здатні розмножуватись, при активізації можуть створити одну копію себе або навіть декілька, а ці копії згодом теж активізуються та створять свої копії і т.д.

1.3. Сфери впливу шкідливих запитів

Існує безліч сфер, які використовують СМО, а отже в кожній із них може виникнути шкідливий запит, який виведе з ладу таку систему або ж завдасть їй шкоди.

СМО використовуються в серверних системах і основним елементом в таких системах виступає сервер. Коли сервер очікує на запит і черга при цьому відсутня, то запит, який надійде, відразу ж подається на сервер і починається оброблятися. Якщо ж сервер зайнятий опрацюванням іншого запиту, то всі запити в такому випадку поміщаються до накопичувача і чекають коли сервер буде готовий до обробки наступного запиту. Після завершення обробки запита сервер відкидає його, задля уникнення повторного його опрацювання. Якщо до системи надходить запит і в цій системі принаймні один сервер вільний, то запит відразу ж відправляється до його сервера. Передбачається, що в системі всі сервери ідентичні, тому якщо до системи надійде запит і буде вільний більше ніж один сервер, то не важливо на який із них вона поступить і який із них буде його обробляти. Сервер в моделі може відображати будь-який пристрій, який виконує якісь функції, надає послуги за відповідними запитами.

Також часто зустрічаються такі системи в промислових підприємствах, де існує транспортне обслуговування, лінії виготовлення продукції, також у

всіх сферах обслуговування, де існують черги, наприклад, вокзали, аеропорти, магазини, банки і т.д., а також сфера обслуговування, кур'єрська доставка, також це може бути комп'ютерна сфера і багато-багато інших сфер, які можуть мати в своїй системі наступні характеристики:

- ймовірність надходження коректного або шкідливого запиту;
- середній час обслуговування запиту;
- середню довжину черги;
- ймовірність втрати запита, який надійшов;
- завантаження обслуговуючих апаратів та інші.

1.4. Різноманіття дій шкідливих запитів

Шкідливі запити можна розділити за деякими ознаками:

- найменш шкідливі – ті, які ніяк не змінюють роботу системи;
- більш шкідливі – ті, які впливають на зменшення пам'яті та спричиняють будь-які ефекти (звукові, графічні і т.д.);
- небезпечні – ті, які можуть спричинити серйозні збої в системі;
- дуже небезпечні – ті, які можуть стерти важливі дані, призвести до видалення програм або ж втрати системи.

Отже, такі запити можуть отримувати доступ до даних системи, змінювати їх, пошкоджувати, викрадати, видаляти, змінювати процеси в системі, а також унеможливлювати роботу самої системи.

З кожним днем шкідливих запитів стає все більше і більше, вони завдають великої шкоди різноманітним компаніям та користувачам, спричиняючи системні критичні помилки, викрадаючи чи модифікуючи приватну інформацію, вражаючи та зупиняючи різноманітні продукти чи системи.

Зараз розберемо типи шкідливого запиту, який може надійти та їх дії:

1. завантажувальний – заражають завантажувальний сектор (boot sector) і можуть спричинити втрату всіх даних, які зберігалися на ньому;

2. файловий – заражають файли:
 - 2.1. файли програм (мають розширення .exe, .com);
 - 2.2. файли з даними деяких програм (наприклад, файли програми Word і т.д.);
 - 2.3. змінюючи і використовуючи їхні імена;
 - 2.4. змінюючи їхню системну інформацію;
3. завантажувально-файловий – можуть заражати як завантажувальний сектор, так і файли;
4. шкідливі запити, які вражають антивірусні програми, роблячи їх непрацездатними або ж некоректно працюючими;
5. шкідливі запити, які фальсифікують зчитану інформацію, надаючи невірні дані користувачу або системі;
6. шкідливі запити, які несуть в своїх заголовках місцезнаходження небезпечних програм, при отриманні такого запиту відбувається активізація цих небезпечних дій.

1.5. Висновки до розділу

В першому розділі кваліфікаційної роботи було розглянуто шкідливі запити, адже вхідний потік розглянутої нами СМО буде складатися не лише з коректних запитів, а й з шкідливих, які будуть надходити до СМО та виводити її канали з ладу. Такі запити є небезпечними, адже вони можуть отримувати доступ до даних системи, змінювати їх, пошкоджувати, викрадати, видаляти, змінювати процеси в системі, а також унеможлиблювати роботу самої системи.

Розглянуто наступні різновиди шкідливих запитів: люки, логічні бомби, троянські коні, запити з черв'яком, зомбі, програмою-викрадачем, поліморфік-генератором. Кожен з них завдає шкоди системам, але відбувається це в різні способи, розпізнати їх можна або на етапі часткового враження, або ж повного, при якому системи виводиться з ладу.

Наведено приклади сфер в яких можуть з'явитися такі запити – загалом це сфери, які в своїх системах мають такі характеристики:

- ймовірність надходження коректного або шкідливого запиту;
- середній час обслуговування запиту;
- середню довжину черги;
- ймовірність втрати запита, який надійшов;
- завантаження обслуговуючих апаратів та інші

Ознайомлено з різноманітністю дій шкідливих запитів за наступними типами:

- завантажувальний;
- файловий;
- завантажувально-файловий;
- вражаючі антивірусні програми;
- фальсифікуючі інформацію;
- з небезпечними заголовками.

РОЗДІЛ 2

АНАЛІТИЧНЕ ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ

Моделювання – це метод вивчення поведінки систем. Він використовується для аналізу якості функціонування систем, прогнозу станів, вибору значень параметрів і структури систем, що забезпечують задані властивості, оптимізацію функціонування моделюючих систем [2].

За своєю суттю моделювання – це метод дослідження системи (оригіналу) за допомогою іншої системи (моделі), що знаходиться в певній відповідності з системою, що вивчається.

Моделювання є одним із способів досліджень систем, який базується на побудові та аналізі системи. В даний час моделюванню систем приділяють значну увагу.

У моделюванні систем важливу роль відіграють моделі СМО. Такі системи дуже часто можна зустріти в повсякденному житті. Особливого значення СМО набули в інформаційній сфері (комп'ютерні системи, бази даних, передавання даних по мережі і т.д.). Трапляються випадки, коли задачу неможливо розв'язати, використовуючи наявні методи в теорії масового обслуговування. Це значно вплинуло на розвиток методів дослідження СМО за допомогою моделювання.

Моделювання дає можливість за допомогою різних моделей аналізувати ситуації в системах, навіть ті, при яких вона може вийти з ладу. Це дає змогу реалізувати критичні ситуації, не причиняючи шкоди реальній системі, й при цьому знаходити їхні найоптимальніші рішення. Основним завданням моделювання є оцінювання показників функціонування системи.

Сучасні програмні засоби для моделювання містять графічний інтерфейс або двовірний чи тривірний простір, що дає змогу краще сприймати

результати моделювання людині, яка не являється спеціалістом в даній області.

Для того, щоб реалізувати достовірну модель потрібно детально ознайомитися з областю застосування системи, методологіями для її моделювання та мовою програмування, якою вона була написана, також потрібно враховувати всі можливі вхідні та вихідні дані і після детального ознайомлення з отриманими позитивними результатами можна розпочинати моделювання. Після отриманих результатів моделювання можливі випадки зміни самої моделі або ж її програмної частини, що може призвести до повторення всіх або ж частини етапів моделювання.

Моделювання складається з декількох етапів:

1 етап - побудова моделі.

2 етап - вивчення, дослідження моделі системи, в тому числі шляхом експериментів на моделі, аналіз результатів моделювання, обробка статистичних даних, які отримані на моделі.

3 етап - перенесення результатів, які отримані за допомогою моделі, на оригінал (систему). За результатами, які були отримані за допомогою моделі, цей етап включає в себе: аналіз якості функціонування системи, прогнозу різного рівня станів системи, вибору значень параметрів і структури системи, що забезпечують задані властивості, оптимізації функціонування моделюючої системи.

Існує багато випадків застосування моделі. Нас будуть цікавити наступні:

- як засіб удосконалення процесів функціонування різних об'єктів;
- як інструмент прогнозування характеристик і показників різних об'єктів і процесів;

- як засіб постановки експериментів для дослідження можливостей удосконалення системи;
- як засіб оптимізації процесу функціонування об'єктів і т.д.

Основний цикл пізнання при моделюванні:

1. Формування незадоволенням існуючого стану справ при функціонуванні системи.
2. Формування шляхів виходу з існуючого стану.
3. Усвідомлення необхідності проведення відповідних досліджень.
4. Формування мети дослідження.
5. На основі аналізу загальних властивостей та детальних властивостей оригіналу здійснюється побудова його моделі.
6. Після побудови моделі в результаті її функціонування отримують результати моделювання, здійснюється статистична обробка цих результатів, за ними обчислюється показники функціонування системи, що моделюється, проводиться їх аналіз.
7. Результати моделювання інтерпретуються у вигляді характеристик та показників оригіналу.
8. За допомогою порівняння здійснюється порівняння (зіставлення) результатів моделювання з результатами, отриманими при безпосередньому дослідженні оригіналу. При цьому порівняння здійснюється в тих самих умовах функціонування.

Якщо результати, отримані за допомогою моделі, в оговореному сенсі близькі до результатів, що отримані при дослідженні оригіналу, то відносно даних властивостей модель можна вважати достатньо адекватною оригіналу.

Розрізняються фізичні та математичні моделі. Фізична модель відрізняється від оригіналу за своєю фізичною природою, проте модель та оригінал підкоряються загальним законам, вони мають різну форму, але характеризуються досить близьким до тотожного математичного опису.

Область застосування фізичних моделей – окремі елементи механічних систем. Фізичні моделі орієнтовані на відтворення фізичних процесів, вони не можуть бути використані при моделюванні дискретних елементів.

При моделюванні інформаційних систем група фізичних моделей не знаходить застосування.

Метод математичного моделювання у порівнянні з фізичним моделюванням має більш широкі можливості. Він є способом дослідження різних об'єктів шляхом складання відповідного математичного опису і обчислення на його основі характеристик і показників функціонування об'єкта, що досліджується.

Суттєвою особливістю даного метода є необхідність побудови математичної моделі. Вона повинна формалізовано відображати процес функціонування оригіналу за допомогою відповідного математичного апарату та описувати основні закономірності його поведінки. При цьому всі другорядні, несуттєві фактори з розгляду виключаються. При цьому, якщо спостерігається деяка суттєва невідповідність між результатами моделювання та результатами оригіналу, що отримані в тих самих умовах експлуатації, за можливістю, проводиться корекція списку основних та другорядних факторів.

Основними поняттями в моделюванні є система та модель.

Система – цілий комплекс, який містить взаємозалежні між собою елементи. Система містить свою унікальну структуру та взаємодіє з зовнішнім середовищем. Система може містити в собі підсистеми або ж бути складовою іншої системи, тобто може формуватися ієрархія з систем [3].

Об'єктом математичного моделювання є складні системи. Складною системою називають певним чином організовану сукупність великої кількості інформаційно зв'язаних та взаємодіючих елементів, що функціонують цілеспрямовано в умовах впливу дії зовнішніх факторів.

Можна виділити два типи задач, що розв'язуються при проектуванні складних систем:

1 тип – задачі аналізу систем (пов’язані з визначенням властивостей, значень параметрів та характеристик системи);

2 тип – задачі синтезу (пов’язані з визначенням складу елементів, їх структури, взаємозв’язків з метою аналізу альтернативних варіантів побудови системи для вибору оптимального).

Обидва класи задач розв’язуються на основі математичного опису складних систем, які представляють собою формалізований опис процесу функціонування системи в часі [4].

Побудова математичного опису складної системи включає в себе наступні етапи:

1. Формалізується мета дослідження.
2. Визначаються параметри, які характеризують параметри системи.
3. Визначають стани системи, що представляють інтерес, виходячи з поставленої мети дослідження.
4. Процес функціонування системи являє собою послідовну зміну станів системи в часі.
5. Для оцінки результатів функціонування системи аналізуються показники якості функціонування (критерії ефективності).

Модель – це система, яка використовується для дослідження іншої системи [5].

Методи математичного моделювання діляться на дві групи:

1. Аналітичні.
2. Імітаційні.

Для того аби визначити види моделей потрібно вказати спосіб подання моделі, розрізняють реальні та абстрактні моделі (рис. 2.1).

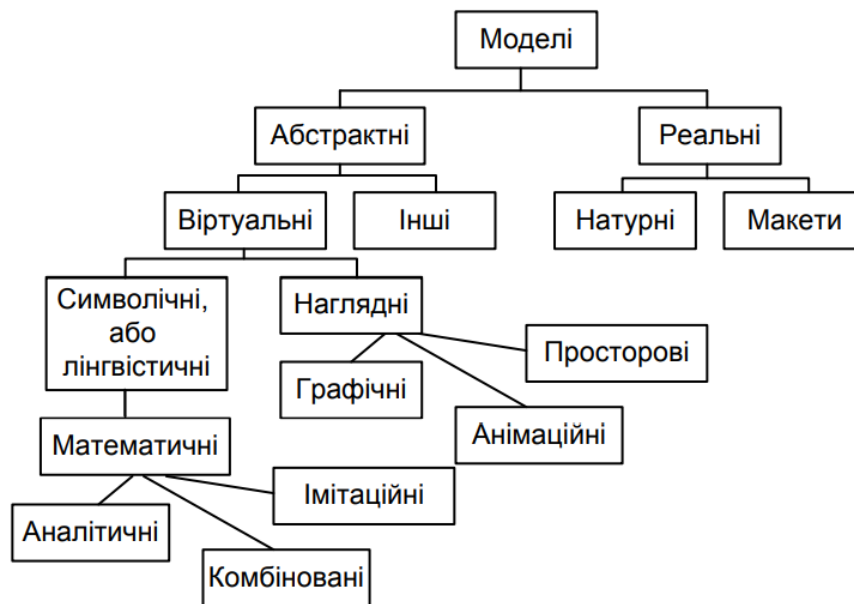


Рис. 2.1. Типи моделей

Під час моделювання основною є абстрактна модель, а основним її видом – математична модель. Ця модель описує реальний об’єкт з максимальною наближеністю до нього. Математична модель поділяється на імітаційну, аналітичну та комбінаційну моделі [6]. В цьому розділі будуть розглянуті наступні моделі теорії масового обслуговування: імітаційна та аналітична.

2.1. Поняття аналітичної моделі

Аналітична модель – це один з видів математичної моделі, який базується на вираженні змін в системі через математичні аналітичні функції. Аналітичну модель використовують для отримання інформації про поведінку системи за допомогою вивчення її математичних функцій, але необхідно володіти математикою на достатньому рівні, щоб зрозуміти суть, яка закладена в цих формулах. Перевага моделі в тому, що не потрібно будувати графіки для отримання інформації про систему, адже це дуже кропіткий та тривалий процес. Недоліком є те, що рішення математично отримати буває важко.

За допомогою цієї моделі робота системи представляється за допомогою логічних умов чи співвідношень.

Методи дослідження аналітичної моделі:

- аналітичний метод – отримання залежностей для шуканих характеристик;
- чисельний метод – отримання числових результатів при відомих початкових даних, не розв'язуючи рівняння;
- якісний метод – отримання властивостей розв'язку, не маючи його в явному вигляді.

Детальне дослідження системи можна провести при наявності залежностей між початковими даними та шуканими характеристиками, які можна отримати в простих системах, а от в складних системах з цим виникають труднощі. Тому при використанні аналітичного методу рекомендується обирати спрощену систему, щоб мати можливість отримати властивості системи. При використанні чисельного методу, на відміну від аналітичного, можна використовувати для дослідження більшу різноманітність систем, але розв'язки будуть приватного типу. Якісний метод використовується якщо результати цілком задовольняють висновки, часто вживаний в теорії автоматичного управління.

Аналітичні методи ґрунтуються на формалізованому описі процесу функціонування об'єкта, коли записуються математичні співвідношення моделі, які потім розв'язуються суцільно математичними методами відносно параметрів, що відшукуються.

Аналітичні методи моделювання при дослідженні складних систем мають обмежене застосування. Це обумовлено тим, що побудова математичних моделей, що мають аналітичний розв'язок, завжди зв'язана з великою кількістю спрощуючих деталей. Причому, підтвердити достовірність цих припущень практично неможливо.

Тому при моделюванні складних систем велике застосування мають численні методи імітаційного моделювання

2.2. Поняття імітаційної моделі

Імітаційна модель – це один з видів математичної моделі, який описує систему в плані логіки, яка може бути використана для проектування, аналізу та оцінки функціонування системи. Ця модель містить мінімальну базову структуру, яку можна розширити відповідно до специфіки задачі [7].

Ці методи дослідження полягають в імітації на моделях, що функціонують поза системою, процесів функціонування системи. Імітуються елементарні явища зі збереженням їх логічної структури, взаємодії та взаємозв'язків, а також послідовності станів у часі.

Оскільки методи імітаційного моделювання реалізуються на електронно-обчислювальних машинах (ЕОМ), математична модель системи повинна бути перетворена у моделюючий алгоритм та програму, яка по фіксованим наборам вихідних даних дозволяє відтворити процес функціонування систем, тобто отримати інформацію про стани системи в певні моменти часу та на її основі обраховувати характеристики якості функціонування системи.

Імітаційну модель можна переглянути в ході процесу, в 2 та 3D форматах. Ця модель дає можливість для проведення аналізу під час виконання моделі, для вивчення процесів, які в ній відбуваються, для внесення змін, що дозволяє проаналізувати її роботу, тим самим прискорити рішення наявного завдання.

За допомогою цієї моделі робота системи представляється в часі, при цьому моделюються явища зі збереженою структурою та проміжком часу, в якому вони протікають, що дає змогу в конкретний час прослідкувати за зміною станів та визначити системні характеристики.

Принципи побудови алгоритмів імітаційного моделювання:

1. Принцип « Δt ».

Він дозволяє у ході імітаційного моделювання отримувати стани системи через фіксовані інтервали часу Δt . При цьому визначається стан системи у початковий момент часу $Z(t_0)$.

Потім, з використанням перетвореного в моделюючу програму математичного описання систем, послідовно визначаються стани системи з кроком Δt : $Z(t_0), Z(t_0 + \Delta t), \dots, Z(t_0 + i\Delta t)$.

Принцип Δt є найбільш універсальним способом побудови моделюючих алгоритмів та використовується в багатьох мовах моделювання (наприклад, GPSS).

Недолік принципу Δt : з точки зору економічності – якщо задається малий крок Δt руху в часі, то при цьому збільшуються затрати машинного часу; при великому кроці Δt можливі ситуації, коли стани, що цікавлять дослідника, можуть бути пропущені.

2. Принцип зміни станів.

При цьому принципі моделюється траєкторія змін станів системи від одного стану до наступного. В цьому випадку рух в часі здійснюється кроками змінної довжини і за рахунок цього скорочуються затрати машинного часу.

3. Принцип послідовного провадження заявок.

Використовується при моделюванні СМО, коли послідовно, фаза за фазою, відтворюється історія проходження заявкою СМО.

Для моделювання інформаційних систем найчастіше використовується зміни станів.

Характерні особливості імітаційного моделювання:

1. За своєю суттю є експериментом (не натурним).
2. Майже немає обмежень на область застосування.
3. Практично є стохастичним експериментом.

Переваги методу імітаційного моделювання:

1. Універсальність методу. Використання у різних предметних областях.

2. Розв'язання задач високої складності, коли аналітичний розв'язок або неможливий, або занадто наближений з великими затратами.

3. Незалежність від фізичної природи процесу, що досліджується, його параметрів.

4. Неможливість аналізу ефективності функціонування систем при різних альтернативних варіантах побудови та вибору оптимального варіанту структури та параметрів системи.

5. У порівнянні з фізичним моделюванням, відсутня необхідність створення фізичного макету системи.

Перевага цієї системи від аналітичної в тому, що за допомогою імітаційної моделі можна відобразити більше даних, тому вона більш точніша, а також, використовуючи імітаційну модель, можна вирішити складні задачі. Аналітичні моделі беруть до уваги різні фактори (присутність безперервних чи дискретних елементів, випадкових дій, нелінійних характеристик і т.д.), які викликають труднощі з використанням аналітичних моделей. Імітаційна модель ефективна при дослідженні складних систем.

Недоліки методу імітаційного моделювання:

1. Необхідність розробки та відлагодження складних алгоритмів та програм на високопродуктивних ЕОМ з великими ресурсами пам'яті.

2. У порівнянні з прямими аналітичними методами, розв'язок, як і при будь-якому чисельному методі, завжди носить частковий характер. Тому для аналізу ефективності функціонування системи потрібно багаторазово відтворювати процес її функціонування при різних наборах вихідних даних. Це призводить до значних витрат обчислювальних ресурсів.

3. Необхідність багаторазово відтворювати процес функціонування системи для аналізу кожного варіанту її побудови і здійснення відповідної статистичної обробки результатів моделювання.

Етапи розробки імітаційної моделі:

1. Побудова концептуальної моделі системи та її формалізація.

2. Алгоритмізація моделі системи та розробка моделюючої програми.

3. Отримання та інтерпретація попередніх результатів моделювання.
4. Перевірка достатньої адекватності моделі та системи.
5. Основний розрахунок показників якості функціонування системи за результатами моделювання.

Основний зміст першого етапу – перехід від змістовного описання до його математичної моделі.

Змістовне описання процесу є першим кроком, в якому треба чітко викласти закономірності, які є характерними для досліджуваного процесу та сформулювати постановку прикладної задачі.

Змістовне описання має відомості про фізичну природу та кількісні характеристики елементарних явищ досліджуваного процесу, про степінь та характер взаємодії між ними, про місце кожного елементарного явища в процесі функціонування системи.

Постановка задачі повинна обов'язково мати чітке викладення ідеї дослідження, список залежностей (за оцінкою результатів моделювання), список факторів (які були враховані при побудові математичної моделі), список показників якості функціонування системи (які визначаються на її моделі), вимоги до допустимого розходження між значенням цих показників для моделі та системи і одних і тих же умовах функціонування.

Формалізована схема процесу є проміжним етапом між змістовним описанням і математичною моделлю.

Формалізована схема повинна бути вже дуже формальним описанням процесу.

Для побудови формалізованої схеми необхідно вибрати характеристики процесу, встановити систему параметрів, що визначають процес, описати всі залежності між характеристиками та параметрами процесу з урахуванням факторів, що були прийняті при формалізації.

Формалізована схема повинна мати точне математичне формулювання задачі дослідження з списком величин, що відшукуються, та оцінених

залежностей. В неї також входять систематизована та уточнена сукупність всіх вихідних даних, параметрів процесу та початкових умов.

Формалізована схема є результатом аналізу та експериментального дослідження процесу. Подальше перетворення формалізованої схеми в математичну модель виконується математичними методами без залучення додаткової інформації про процес.

Формалізована схема розроблюється сумісними зусиллями спеціалістів відповідної прикладної області знань та математиків.

Для перетворення формалізованої схеми в математичну модель необхідно записати в аналітичній формі всі співвідношення, виразити логічні умови у вигляді систем нерівностей та придати аналітичну форму всім відомостям, що є у формалізованій схемі (таблиці, графіки замінюються при необхідності інтерполяційними поліномами і т.п.).

Перший етап моделювання закінчується розробкою загальної блок-схеми математичної моделі.

На другому етапі моделювання – етапі алгоритмізації та її програмної реалізації, математична модель перетворюється у конкретну машинну модель. При цьому можуть складатися узагальнені та детальні логічні схеми моделюючих алгоритмів та блок-схеми програм.

Узагальнена блок-схема алгоритму описує порядок дій при моделюванні систем без будь-яких деталей.

Описання узагальненої схеми моделюючого алгоритму:

1. Інформаційні системи з достатньою ступенем адекватності можуть бути формалізовані у вигляді систем або мереж масового обслуговування. Для систем та мереж масового обслуговування характерною особливістю є наявність великої кількості випадкових величин (ВВ), що мають різні функції розподілу.

В силу цього велике значення має генерація в ЕОМ ВВ з заданими функціями розподілу. Тому в узагальненій схемі моделюючого алгоритму

першим блоком є датчик випадкових чисел (ДВЧ), що генерує випадкові числа, що рівномірно розподілені в інтервалі $[0,1]$.

2. Наступним блоком є блок генерації ВВ з заданими законами розподілу. Ці випадкові числа беруть участь як у формуванні вхідного потоку повідомлень, так і в обробці та можливих передислокаціях. Ці два блоки є в певному сенсі підготовчими, проте вони повинні бути присутніми в моделях інформаційних систем.

3. Наступним блоком є блок, що здійснює функції оператора обчислення показників ефективності за реалізаціями моделі за детальною схемою моделюючого алгоритму. Цей блок складає серцевину моделі, її саму суттєву частину. Тому цей блок повинен бути пунктуально пророблений, так як від нього у значній мірі залежить якість моделі, тобто степінь її адекватності об'єкту – інформаційній системі. Начинкою цього блоку є формалізація процесу надходження і алгоритмів обробки повідомлень, які надходять.

4. Обчислені показники ефективності за реалізаціями моделі надходять до блоку – накопичувач реалізацій показників ефективності.

5. Далі йде блок обробки результатів моделювання. Блок в залежності від потреб повинен реалізовувати весь арсенал математичної статистики. Сюди входить: побудова гістограм, емпіричних функцій розподілу і т.д.

6. Всі обчислення йдуть за циклом від однієї реалізації до наступної.

Детальна схема моделюючого алгоритму має уточнення, які відсутні в узагальненій схемі, вона показує не тільки те, що потрібно виконувати на черговому кроці моделювання, а й саму процедуру виконання даного кроку.

Логічна схема моделюючого алгоритму вказує впорядковану в часі послідовність логічних операцій, що пов'язані з розв'язанням задачі моделювання.

Блок-схема програми являє собою інтерпретацію логічної схеми моделюючого алгоритму у вигляді програми на базі конкретної алгоритмічної мови.

На етапі алгоритмізації та програмної реалізації моделі виконуються перевірки:

- адекватності логічної схеми моделюючого алгоритму логічній схемі моделі;
- окремих частин програми – на прикладах розв’язання різних тестових задач;
- всієї програми – на контрольному прикладі моделювання системи.

На третьому етапі моделювання – етапі отримання та попередньої інтерпретації результатів моделювання – ЕОМ використовується для проведення машинних експериментів з використанням складеної програми моделювання.

Слід зазначити, що необхідно планувати машинні експерименти. Про планування машинних експериментів написано ряд книг, наприклад, Налімов В.В. Теорія експерименту, 1971, М.:Наука (рос), Налімов В.В. Статистичні методи планування екстремальних експериментів. М.:Наука (рос).

Отримавши та проаналізувавши результати моделювання, їх необхідно інтерпретувати по відношенню до об’єкта моделювання, тобто зробити перехід від інформації, яка отримана в результаті машинного експерименту з моделлю до інформації про об’єкт моделювання, на основі якої і будуть робитися висновки відносно характеристик процесу функціонування системи, яка досліджується.

На четвертому етапі моделювання – перевірки адекватності моделі та системи - перевіряється виконання вимог до допустимого розходження між значеннями цих показників для моделі та системи в тих самих умовах функціонування. Якщо вимоги виконуються, то модель вважається достатньо адекватною.

На п'ятому, завершальному, етапі моделювання здійснюється основний розрахунок показників якості функціонування системи за результатами моделювання.

Імітаційна модель утворюється взаємодією елементів: станів, подій, датчиків, випадкових чисел, таймера, ланцюгів подій, мети моделювання, лічильника, блоку ініціалізації, критерія зупинки, методів обробки результатів [8].

Стан системи повинен бути визначений зі ступеню деталізації, яка необхідна та достатня для того, щоб процес зміни стану моделі в часі був Марківським.

Подія моделі розуміється як стрибова зміна її стану (надходження вимоги, закінчення його обслуговування і т.д.).

За допомогою датчика випадкових чисел генерується чергова випадкова величина зі своїми розподілами, за якими формуються чергові стани.

Процес, що імітується, розвивається у модельному часі. Лічильник модельного часу є таймером.

Основною частиною моделі є Марківський процес, який і визначає всі операції. Які здійснюються в моделі в рамках її реалізованого алгоритму.

Введення Марківського процесу є одним з найважливіших елементів побудови моделі інформаційної системи.

По суті, функціонування імітаційної моделі у модельному часі є зміна станів у цьому часі введеного Марківського процесу.

Логіка моделі реалізується в процесі реалізації ланцюга подій (вихід обслуженої вимоги, зв'язане з виходом просування черги і т.д.).

Ініціалізація – це приведення моделі до початку прогону у вихідний стан (обнулінням всіх накопичувачів лічильників, встановлення у вихідний стан генераторів псевдовипадкових чисел).

Вибір мети визначають лічильники, необхідні для накопичування результатів моделювання.

Якщо мета – визначення розподілу числа вимог у системі, то потрібний лічильник поточного числа вимог у системі.

Якщо мета – визначення розподілу часу очікування вимог в системі, то потрібний лічильник часів очікування вимог, що виходять з системи і т.д.

Показники якості функціонування моделюючої системи часто потрібні у стаціонарному режимі (при спрямуванні до нескінченності модельного часу). Дані, що накопичені за час перехідного до такого режиму процесу, будуть вносити помилку в кінцеві результати. Щоб цього не було, результати перехідного режиму треба відкинути. При необхідності декількох прогонів для отримання показників якості функціонування системи, що моделюється. В стаціонарному режимі це відкидання стає досить непродуктивним. В таких випадках для ергодичних процесів, стаціонарний режим яких не залежить від початкових умов, беруть одну досить довгу реалізацію моделі.

Критерій зупинки визначає момент припинення прогону моделі. У найпростішому випадку прогон закінчується як тільки буде досягнуте задане значення таймера, лічильника числа обслужених вимог і т.д. Проте вірніше управляти прогоном як тільки буде досягнута задана точність показників, що визначаються. Розумний вибір правила зупини нетривіальний, так як на етапі планування експериментів на моделі ні оцінка величини, що визначається, ні її дисперсія, як правило, не відомі.

Обробка результатів моделювання складається із стиснення отриманої інформації, обчисленні оцінок математичних сподівань та вищих моментів для випадкових величин, статистичної оцінки середніх, побудові гістограм та статистичних розподілів, а також відповідних регресійних моделей. Відповідно, в систему моделювання повинні включатися необхідні підпрограми широкого застосування, які слабко зв'язані за специфікою моделі.

2.3. Висновки до розділу

Моделювання є одним із способів досліджень систем, який базується на побудові та аналізі системи. В даний час моделюванню систем приділяють значну увагу.

Моделювання дає можливість за допомогою різних моделей аналізувати ситуації в системах, навіть ті, при яких вона може вийти з ладу. Це дає змогу реалізувати критичні ситуації, не причиняючи шкоди реальній системі, й при цьому знаходити їхні найоптимальніші рішення. Основним завданням моделювання є оцінювання показників функціонування системи.

В другому розділі кваліфікаційної роботи було розглянуто три етапи моделювання систем, основні поняття в моделюванні, ознайомлено з моделями моделювання, розглянуті наступні моделі: імітаційну та аналітичну, наведено їхні переваги та недоліки використання, методи їх дослідження, принципи побудови та етапи розробки.

РОЗДІЛ 3

АНАЛІТИЧНА МОДЕЛЬ СМО З ШКІДЛИВИМИ ЗАПИТАМИ

На n -канальну систему масового обслуговування надходить Пуассонівський потік заявок, утворений підпотоками коректних запитів на обслуговування і шкідливих запитів з параметрами λ , $a\lambda$, $(1-a)\lambda$ відповідно, $a > 0.5$. Обслуговування коректних запитів експоненціальне з параметром μ . Шкідливий запит, що надійшов до системи, блокує канал, виводячи його з обслуговування, займає його безкінечно довго при увімкненій системі. Відмова системи настає з першою втратою коректного запиту, який надійшов на вимкнену систему, це момент виявлення відмови. Відразу починається відновлення виведених зі строю каналів обслуговування по експоненціальному закону з параметром ν .

Можна виділити два режими роботи системи:

- 1) система увімкнена, якщо система вільна або обслуговує коректний запит;
- 2) система вимкнена, якщо надійшов шкідливий запит.

Опишемо роботу системи, яка підлягає ремонту з n обслуговуючими каналами, на які надходять коректні та шкідливі запити. Побудуємо графі станів з інтенсивностями переходів. Відмітимо марковість процесу зміни стану. Обчислимо P_{ij} – ймовірності переходів із стану i в стан j . Побудуємо матрицю переходів $P = (P_{ij})$. Випишемо систему рівнянь Чепмена-

Колмогорова для $P'_i(t)$ і знайдемо показники ефективності досліджуваної моделі СМО.

3.1. Одноканальна СМО

Одноканальна СМО – це система масового обслуговування, яка має лише один канал [9]. На цей канал надходять запити з інтенсивністю λ і якщо канал вільний, то запит починає оброблятися з інтенсивністю μ , якщо ж канал зайнятий, то утворюється черга запитів, де кожен новий запит стає в кінець, а коли канал звільняється, то на нього надходить перший запит із цієї черги, якщо вона наявна, а якщо черга не передбачена, то запит, який надійде буде відхилений [10].

Розглядаємо випадок, коли $n = 1$. $S_i(x, y)$ – стан системи, де x – кількість шкідливих запитів, y – кількість коректних запитів.

Така система може знаходитися в чотирьох станах – це очікування запиту S_0 , обслуговування коректного запиту S_1 , обслуговування шкідливого запиту S_2 та ремонт S_3 .

$S_0 = (0,0)$ – 0 шкідливих запитів, 0 коректних запитів,

$S_1 = (0,1)$ – 0 шкідливих запитів, 1 коректний запит,

$S_2 = (1,0)$ – 1 шкідливий запит, 0 коректних запитів,

S_3 – ремонт системи.

Позначимо ймовірності станів:

$P_0(t)$ - ймовірність стану очікування запиту (канал вільний),

$P_1(t)$ - ймовірність стану обслуговування коректного запиту (канал зайнятий),

$P_2(t)$ - ймовірність стану обслуговування шкідливого запиту (канал зайнятий),

$P_3(t)$ - ймовірність стану ремонту.

Нехай $a\lambda$ – потік неприйнятих коректних запитів з інтенсивністю $a\lambda$.

Граф станів з інтенсивністю переходів для СМО з одним каналом обслуговування зображено на рисунку 3.1.

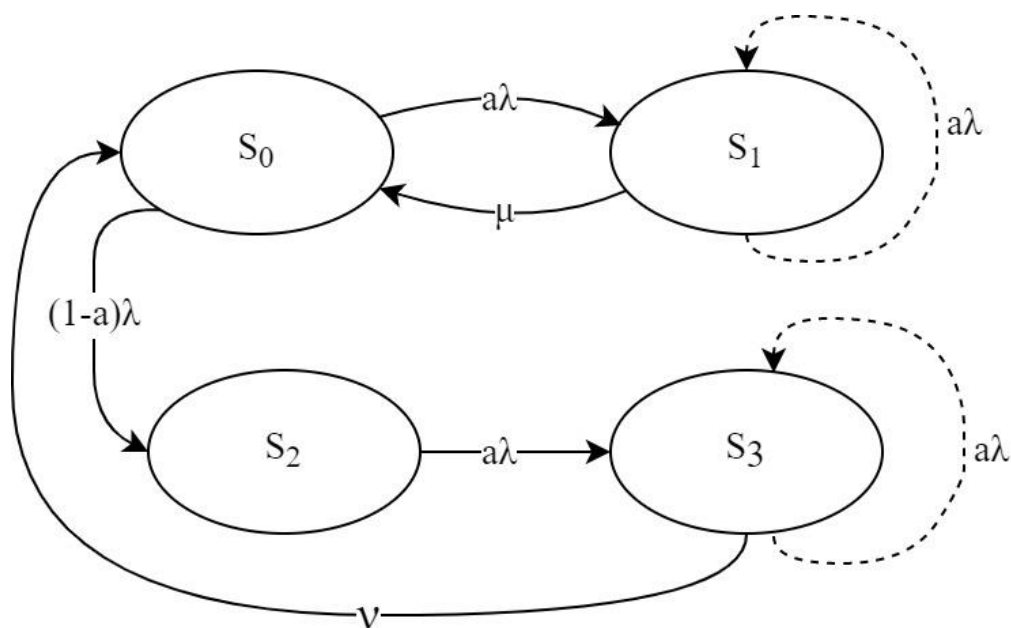


Рис. 3.1. Граф станів з інтенсивністю переходів для СМО з одним каналом обслуговування

Розрахуємо ймовірності переходів по станах:

1) для стану S_0 :

Оскільки з стану S_0 можна перейти до стану S_1 та стану S_2 , то можна розрахувати ймовірність переходу системи з стану S_0 до стану S_1 - P_{01} та ймовірність переходу системи з стану S_0 до стану S_2 - P_{02} :

- ймовірність переходу стану S_0 в S_1 :

Ймовірність того, що стан системи S_0 перейде в стан S_1 раніше, ніж в S_2 :

$$P_{01} = P((S_0 \rightarrow S_1) < (S_0 \rightarrow S_2))$$

$$\left\{ \begin{array}{l} (S_0 \rightarrow S_1) - (\xi, F, f) \\ (S_0 \rightarrow S_2) - (\eta, G, g) \\ F(t) = 1 - e^{-a\lambda t} \\ f(t) = a\lambda e^{-a\lambda t} \\ G(t) = 1 - e^{-(1-a)\lambda t} \\ g(t) = (1-a)\lambda e^{-(1-a)\lambda t} \\ \xi, \eta \in (0; +\infty) \end{array} \right.$$

$$P(\xi < \eta) = P(\eta \in (t; t+dt); \xi < t; t \in [0; +\infty)) = \int_0^{+\infty} F(t)g(t)dt,$$

отже, ймовірність переходу стану S_0 в стан S_1 :

$$P_{01} = \int_0^{+\infty} (1 - e^{-a\lambda t})((1-a)\lambda e^{-(1-a)\lambda t}) = a.$$

- ймовірність переходу S_0 в S_2 :

Оскільки ймовірність переходу стану S_0 в стан S_1 - $P_{01} = a$, як відомо, що сума ймовірностей дорівнює 1, то ймовірність переходу стану S_0 в S_2 дорівнює

$$P_{02} = 1 - P_{01} = 1 - a.$$

2) для стану S_1 :

Оскільки зі стану S_1 можна перейти лише в стан S_0 , то ймовірність переходу стану S_1 в стан S_0 - $P_{10} = 1$.

3) для стану S_2 :

Оскільки зі стану S_2 можна перейти лише в стан S_3 , то ймовірність переходу стану S_2 в стан S_3 - $P_{23} = 1$.

4) Для стану S_3 :

Оскільки зі стану S_3 можна перейти лише в стан S_0 , то ймовірність переходу стану S_3 в стан S_0 - $P_{30} = 1$.

В момент зміни станів ймовірності переходів P_{ij} залежать від номерів станів. Тобто вони мають Марківську властивість і процес має вкладений кінцевий ланцюг Маркова.

За допомогою отриманих ймовірностей можемо побудувати матрицю переходів:

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 0 & a & 1-a & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

При $N = 6 \Rightarrow$ матриця $P^N = P^6$ - транзитивна.

Сформуємо систему рівнянь Чепмена-Колмогорова до даної СМО:

$$P_i(t + \Delta t) = \sum_k P_k(t) P_{ki}(\Delta t); \quad i = \overline{0, 3}$$

$$\begin{cases} P_0(t + \Delta t) = P_0(t)P_{00}(\Delta t) + P_1(t)P_{10}(\Delta t) + P_3(t)P_{30}(\Delta t) \\ P_1(t + \Delta t) = P_0(t)P_{01}(\Delta t) + P_1(t)P_{11}(\Delta t) \\ P_2(t + \Delta t) = P_0(t)P_{02}(\Delta t) + P_2(t)P_{22}(\Delta t) \\ P_3(t + \Delta t) = P_2(t)P_{23}(\Delta t) + P_3(t)P_{33}(\Delta t) \end{cases}$$

Підставляємо розраховані ймовірності в систему рівнянь:

$$\begin{cases} P_0(t + \Delta t) = P_0(t)(1 - a\lambda\Delta t) + P_1(t)\mu\Delta t + P_3(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) = P_0(t)a\lambda\Delta t + P_1(t)(1 - \mu\Delta t) + o(\Delta t) \\ P_2(t + \Delta t) = P_0(t)(1 - a)\lambda\Delta t + P_2(t)(1 - a\lambda\Delta t) + o(\Delta t) \\ P_3(t + \Delta t) = P_2(t)a\lambda\Delta t + P_3(t)(1 - \nu\Delta t) + o(\Delta t) \end{cases}$$

$$\begin{cases} P_0(t + \Delta t) = P_0(t) - P_0(t)a\lambda\Delta t + P_1(t)\mu\Delta t + P_3(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) = P_0(t)a\lambda\Delta t + P_1(t) - P_1(t)\mu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) = P_0(t)(1 - a)\lambda\Delta t + P_2(t) - P_2(t)a\lambda\Delta t + o(\Delta t) \\ P_3(t + \Delta t) = P_2(t)a\lambda\Delta t + P_3(t) - P_3(t)\nu\Delta t + o(\Delta t) \end{cases}$$

В правій частині рівнянь залишимо елементи, які мають Δt , а в ліву частину перенесемо інші, які не мають Δt :

$$\begin{cases} P_0(t + \Delta t) - P_0(t) = -P_0(t)a\lambda\Delta t + P_1(t)\mu\Delta t + P_3(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) - P_1(t) = P_0(t)a\lambda\Delta t - P_1(t)\mu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) - P_2(t) = P_0(t)(1 - a)\lambda\Delta t - P_2(t)a\lambda\Delta t + o(\Delta t) \\ P_3(t + \Delta t) - P_3(t) = P_2(t)a\lambda\Delta t - P_3(t)\nu\Delta t + o(\Delta t) \end{cases}$$

$$\begin{cases} \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} = -P_0(t)a\lambda + P_1(t)\mu + P_3(t)\nu \\ \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = P_0(t)a\lambda - P_1(t)\mu \\ \frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} = P_0(t)(1 - a)\lambda - P_2(t)a\lambda \\ \frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} = P_2(t)a\lambda - P_3(t)\nu \end{cases}$$

При $\Delta t \rightarrow 0$, границі правих частин рівнянь існують, тож вони існують і зліва, тоді це похідні функції $P_i(t)$, $i=\overline{0,3}$. Тоді запишемо систему диференціальних рівнянь:

$$\begin{cases} P_0'(t) = -P_0(t)a\lambda + P_1(t)\mu + P_3(t)\nu \\ P_1'(t) = P_0(t)a\lambda - P_1(t)\mu \\ P_2'(t) = P_0(t)(1 - a)\lambda - P_2(t)a\lambda \\ P_3'(t) = P_2(t)a\lambda - P_3(t)\nu \end{cases}$$

Маємо систему чотирьох лінійних диференціальних рівнянь першого порядку з постійними коефіцієнтами. Задамо початкові умови, щоб вирішити систему рівнянь Колмогорова й знайти ймовірності станів:

$$P_0(0) = 1;$$

$$P_1(0) = 0;$$

$$P_2(0) = 0;$$

$$P_3(0) = 0.$$

Для вирішення приведеної системи рівнянь використаємо математичний пакет Matlab. За допомогою команди `sums` оголошуємо змінні

(рис. 3.2), які використовуються в цій системі та використовуємо функцію `dsolve()` [11] для вирішення диференціальних рівнянь (рис. 3.3).

```
>> %Оголошуємо змінні
>> %де а - а, l - лямбда, t - час, n - ню, m - мя
>> syms a l t n m;
>>
```

Рис. 3.2. Оголошення змінних

```
>> %Вирішуємо систему диференціальних рівнянь
>> dsolve('Dx0(t)=-x0(t)*a+l+x1(t)*m+x3(t)*n', 'Dx1(t)=x0(t)*a+l-x1(t)*m', 'Dx2(t)=x0(t)*(1-a)*l-x2(t)*a+l',
'Dx3(t)=x2(t)*a+l-x3(t)*n', 'x0(0)=1', 'x1(0)=0', 'x2(0)=0', 'x3(0)=0')
```

Рис. 3.3. Вирішення диференціальних рівнянь

Отримуємо результати, які представлені на рисунках 3.4 – 3.6.

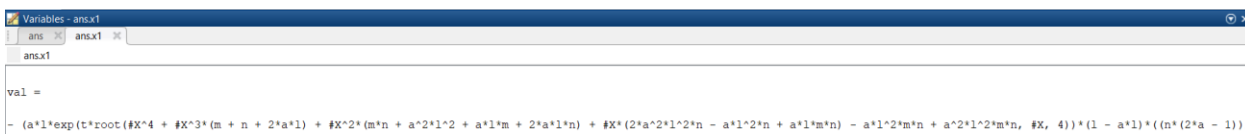


Рис. 3.4. Обраховане значення x_0

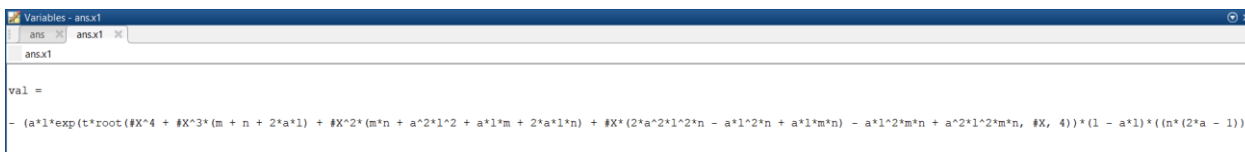


Рис. 3.5. Обраховане значення x_1

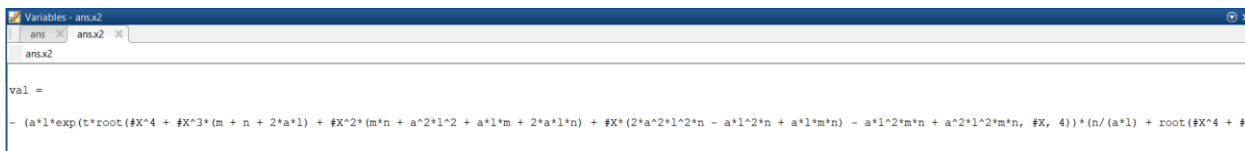


Рис. 3.6. Обраховане значення x_2

```

Variables - ansx3
ans - ansx3
ansx3

val =
- (a^1*exp(t*root(#X^4 + #X^3*(m + n + 2*a^1) + #X^2*(m*n + a^2*1^2 + a^1*m + 2*a^1*n) + #X*(2*a^2*1^2*n - a^1*2*n + a^1*m*n) - a^1*2*m*n + a^2*1^2*m*n, #X, 4))^(1 - a^1)*(root(#X^4 + #X

```

Рис. 3.7. Обраховане значення х3

Рішення даної системи диференційних рівнянь:

$$P_0(t) = P_0(a, \lambda, t, \nu, \mu);$$

$$P_1(t) = P_1(a, \lambda, t, \nu, \mu);$$

$$P_2(t) = P_2(a, \lambda, t, \nu, \mu);$$

$$P_3(t) = P_3(a, \lambda, t, \nu, \mu).$$

Знайдемо остаточні ймовірності:

$$\lim_{t \rightarrow \infty} P_i(t) = P_i; \quad i = \overline{0,3}; \quad \sum_{i=0}^3 P_i = 1.$$

Праві частини приведених рівнянь системи стають постійними, а, отже, й ліві теж постійні. Ці значення мають бути нулями, оскільки в іншому випадку $\lim_{t \rightarrow \infty} P_i(t)$ вийде з області допустимих значень $[0;1]$, що недопустимо.

Тобто для знаходження ймовірностей прирівнюємо ліві частини системи рівнянь Колмогорова до нуля й вирішуємо тепер не диференціальну систему рівнянь, а лінійну.

$$\begin{cases} 0 = -P_0 a \lambda + P_1 \mu + P_3 \nu \\ 0 = P_0 a \lambda - P_1 \mu \\ 0 = P_0 (1 - a) \lambda - P_2 a \lambda \\ 0 = P_2 a \lambda - P_3 \nu \end{cases}$$

Знову використаємо математичний пакет Matlab для знаходження розв'язку системи рівнянь. За допомогою команди `syms` оголошуємо змінні (рис. 3.8) та використовуємо функцію `solve()` [13] для вирішення лінійних алгебраїчних рівнянь (рис. 3.9).

```

>> %Оголошуємо змінні
>> syms x0 x1 x2 x3;
>>

```

Рис. 3.8. Оголошення змінних

```

>> x1 = solve(0 == x0*a*1 - x1*m, x1)

x1 =

(a*1*x0)/m

>> x2 = solve(0 == x0*(1-a)*1 - x2*a*1, x2)

x2 =

-(x0*(a - 1))/a

>> x3 = solve(0 == x2*a*1 - x3*n, x3)

x3 =

-(1*x0*(a - 1))/n

>> x0 = 1 - x1 - x2 - x3

x0 =

(x0*(a - 1))/a - (a*1*x0)/m + (1*x0*(a - 1))/n + 1

>> x0 = solve(x0)

x0 =

-1/((a - 1)/a + (1*(a - 1))/n - (a*1)/m)

```

Рис. 3.9. Вирішення лінійних алгебраїчних рівнянь

Отримуємо рішення рівнянь у вигляді:

$$P_0(t) = P_0(a, \lambda, \nu, \mu);$$

$$P_1(t) = P_1(a, \lambda, \nu, \mu);$$

$$P_2(t) = P_2(a, \lambda, \nu, \mu);$$

$$P_3(t) = P_3(a, \lambda, \nu, \mu).$$

Також можна аналітично вирішити цю систему рівнянь, виразивши кожне P_i , де $i=1,2,3$ через P_0 :

$$\begin{cases} P_0 a \lambda = P_1 \mu + P_3 \nu \\ P_1 \mu = P_0 a \lambda \\ P_2 a \lambda = P_0 (1 - a) \lambda \\ P_3 \nu = P_2 a \lambda \end{cases}$$

$$P_1 = \frac{a\lambda}{\mu} P_0;$$

(1.1)

$$P_2 = \frac{(1-a)\lambda}{a\lambda} P_0 = \frac{1-a}{a} P_0;$$

(1.2)

$$P_3 = \frac{a\lambda}{\nu} P_2 = \frac{(1-a)a\lambda}{a\nu} P_0 = \frac{(1-a)\lambda}{\nu} P_0;$$

(1.3)

Оскільки $P_0 + P_1 + P_2 + P_3 = 1$, то

$$P_0 = 1 - P_1 - P_2 - P_3;$$

$$P_0 = 1 - \frac{a\lambda}{\mu} P_0 - \frac{1-a}{a} P_0 - \frac{(1-a)\lambda}{\nu} P_0;$$

$$P_0 = 1 - P_0 \left(\frac{a\lambda}{\mu} + \frac{1-a}{a} + \frac{(1-a)\lambda}{\nu} \right);$$

$$P_0 + P_0 \left(\frac{a\lambda}{\mu} + \frac{1-a}{a} + \frac{(1-a)\lambda}{\nu} \right) = 1;$$

$$P_0 \left(1 + \frac{a\lambda}{\mu} + \frac{1-a}{a} + \frac{(1-a)\lambda}{\nu} \right) = 1;$$

$$P_0 = \frac{1}{1 + \frac{a\lambda}{\mu} + \frac{1-a}{a} + \frac{(1-a)\lambda}{\nu}};$$

$$P_0 = \frac{1}{\frac{a\mu\nu}{a\mu\nu} + \frac{a^2\lambda\nu}{a\mu\nu} + \frac{(1-a)\mu\nu}{a\mu\nu} + \frac{(1-a)a\lambda\mu}{a\mu\nu}};$$

$$P_0 = \frac{1}{\frac{a\mu\nu + a^2\lambda\nu + (1-a)\mu\nu + (1-a)a\lambda\mu}{a\mu\nu}};$$

$$P_0 = \frac{a\mu\nu}{a\mu\nu + a^2\lambda\nu + (1-a)\mu\nu + (1-a)a\lambda\mu};$$

$$P_0 = \frac{a\mu\nu}{a\mu\nu + a^2\lambda\nu + \mu\nu - a\mu\nu + a\lambda\mu - a^2\lambda\mu};$$

$$P_0 = \frac{a\mu\nu}{a^2\lambda\nu + \mu\nu + a\lambda\mu - a^2\lambda\mu} \tag{1.4}$$

В стаціонарному режимі система знаходиться в S_0, S_1, S_2 та S_3 станах з ймовірностями P_0, P_1, P_2, P_3 відповідно.

Розрахуємо показники ефективності даної СМО:

- 1) ймовірність відмови в обслуговуванні:

$$P_{vidm} = P_n = P_1,$$

де n – кількість каналів в СМО

- 2) ймовірність обслуговування запита:

$$P_{obsl} = 1 - P_{vidm} = 1 - P_1$$

- 3) відносна пропускна здатність:

$$Q = 1 - P_{vidm} = 1 - P_1$$

(1.5)

- 4) абсолютна пропускна здатність:

$$A = a\lambda Q$$

(1.6)

- 5) ймовірність втрати запита (СМО зайнята):

$$P_{vtr} = P_1 + P_2 + P_3 = 1 - P_0$$

(1.7)

- 6) інтенсивність потоку втрачених запитів (кількість запитів, які отримали відмову з одиницю часу)

$$N_{vtr} = a\lambda P_{vtr} = a\lambda(1 - P_0)$$

(1.8)

3.2. Двоканальна СМО

Двоканальна СМО – це багатоканальна ($n > 1$) система масового обслуговування, яка має два канали. Така система може обслуговувати одночасно не більше 2 запитів [14]. На цей канал надходять запити з інтенсивністю λ , потік обслуговування кожного з двох каналів має інтенсивність μ , а середня тривалість обслуговування одного запиту дорівнює $\frac{1}{\mu}$, причому тривалість обслуговування запиту кожним з двох каналів є випадковою величиною(ВВ), підпорядкованою експоненціальному

закону розподілу. Якщо до системи надходить запит і жоден з каналів не зайнятий, то запит починає оброблятися одним із них, якщо ж один з каналів зайнятий, а інший ні, то запит надходить до нього, якщо ж два канали зайняті, то цей запит стає в чергу і чекає коли звільниться один з наявних каналів.

Перевага використання такої системи з двома паралельно під'єднаними каналами (в порівнянні з одноканальною системою) полягає в більш швидкому обслуговуванні запитів за рахунок обслуговування двох запитів одночасно.

Розглядаємо випадок, коли $n = 2$. $S_i(r, x, y)$ – стан системи, де r – кількість каналів, які знаходяться на ремонті, x – кількість шкідливих запитів, y – кількість коректних запитів. Ремонт одного або двох каналів починається при втраті коректного запиту, у випадку коли система зайнята двома шкідливими запитами або одним шкідливим, одною коректною, відбувається виявлення несправності системи.

Дана система може знаходитися в десятих станах:

$S_0 = (0,0,0)$ – 0 каналів на ремонті, 0 шкідливих запитів, 0 коректних запитів (очікування запиту) ,

$S_1 = (0,0,1)$ – 0 каналів на ремонті, 0 шкідливих запитів, 1 коректний запит,

$S_2 = (0,1,0)$ – 0 каналів на ремонті, 1 шкідливий запит, 0 коректних запитів,

$S_3 = (0,1,1)$ – 0 каналів на ремонті, 1 шкідливий запит, 1 коректний запит,

$S_4 = (0,0,2)$ – 0 каналів на ремонті, 0 шкідливих запитів, 2 коректних запиту,

$S_5 = (0,2,0)$ – 0 каналів на ремонті, 2 шкідливих запиту, 0 коректних запитів,

$S_6 = (1,0,0)$ – 1 канал на ремонті, 0 шкідливих запитів, 0 коректних запитів,

$S_7 = (1,0,1)$ – 1 канал на ремонті, 0 шкідливих запитів, 1 коректний запит,

$S_8 = (1,1,0)$ – 1 канал на ремонті, 1 шкідливий запит, 0 коректних запитів,

$S_9 = (2,0,0)$ – 2 канали на ремонті, 0 шкідливих запитів, 0 коректних запитів.

Позначимо ймовірності станів:

$P_0(t)$ - ймовірність стану очікування запиту (2 канали вільні),

$P_1(t)$ - ймовірність стану обслуговування коректного запиту (1 канал зайнятий),

$P_2(t)$ - ймовірність стану обслуговування шкідливого запиту (1 канал зайнятий),

$P_3(t)$ - ймовірність стану обслуговування шкідливого та коректного запитів (2 канали зайняті),

$P_4(t)$ - ймовірність стану обслуговування коректних запитів (2 канали зайняті),

$P_5(t)$ - ймовірність стану обслуговування шкідливих запитів (2 канали зайняті),

$P_6(t)$ - ймовірність стану ремонту одного з каналів (1 канал на ремонті, 1 канал вільний),

$P_7(t)$ - ймовірність стану ремонту одного з каналів та обслуговування коректного запиту іншим (1 канал на ремонті, 1 канал зайнятий),

$P_8(t)$ - ймовірність стану ремонту одного з каналів та обслуговування шкідливого запиту іншим (1 канал на ремонті, 1 канал зайнятий),

$P_9(t)$ - ймовірність стану ремонту двох каналів.

Нехай $a\lambda$ – потік неприйнятих коректних запитів з інтенсивністю $a\lambda$.

Граф станів з інтенсивністю переходів для СМО з двома каналами обслуговування зображено на рисунку 3.10.

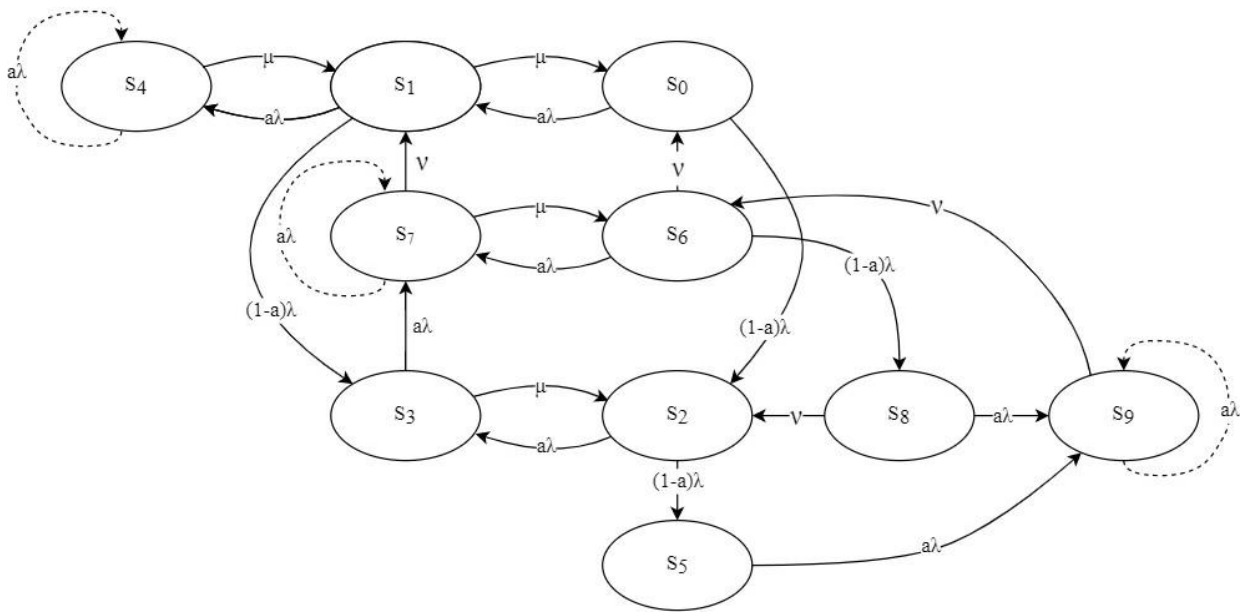


Рис. 3.10. Граф станів з інтенсивністю переходів для СМО з двома каналами обслуговування

Розрахуємо ймовірності переходів по станах:

1) для стану S_0 :

Оскільки з стану S_0 можна перейти до стану S_1 та стану S_2 , то можна розрахувати ймовірність переходу системи з стану S_0 до стану S_1 - P_{01} та ймовірність переходу системи з стану S_0 до стану S_2 - P_{02} :

- ймовірність переходу стану S_0 в S_1
 $P_{01} = a$ (розраховано в попередньому пункті).
- ймовірність переходу S_0 в S_2

Оскільки ймовірність переходу стану S_0 в стан S_1 - $P_{01} = a$, як відомо, що сума ймовірностей дорівнює 1, то ймовірність переходу стану S_0 в S_2 дорівнює

$$P_{02} = 1 - P_{01} = 1 - a.$$

2) для стану S_1 :

Оскільки з стану S_1 можна перейти до стану S_0 , S_3 , та стану S_4 , то можна розрахувати ймовірність переходу системи з стану S_1 до стану S_0 -

P_{10} , ймовірність переходу системи з стану S_1 до стану S_3 - P_{13} та ймовірність переходу системи з стану S_1 до стану S_4 - P_{14} :

- ймовірність переходу стану S_1 в S_0 :

$$\text{Оскільки } P_0 = \frac{1}{\mu+\lambda}, \text{ то } P_{10} = \mu \cdot P_0 = \mu \cdot \frac{1}{\mu+\lambda} = \frac{\mu}{\mu+\lambda}.$$

- ймовірність переходу стану S_1 в S_3 :

$$P_{13} = (1-a) \cdot \lambda \cdot P_0 = (1-a) \cdot \lambda \cdot \frac{1}{\mu+\lambda} = \frac{(1-a)\lambda}{\mu+\lambda}.$$

- ймовірність переходу стану S_1 в S_4 :

$$P_{14} = a \cdot \lambda \cdot P_0 = a \cdot \lambda \cdot \frac{1}{\mu+\lambda} = \frac{a\lambda}{\mu+\lambda}.$$

3) для стану S_2 :

Оскільки зі стану S_2 можна перейти до стану S_3 та S_5 , то можна розрахувати ймовірність переходу системи з стану S_2 до стану S_3 - P_{23} та ймовірність переходу системи з стану S_2 до стану S_5 - P_{25} :

- ймовірність переходу стану S_2 в S_3 :

$$P_{23} = (1 - a).$$

- ймовірність переходу стану S_2 в S_5 :

$$P_{25} = a.$$

4) для стану S_3 :

Оскільки зі стану S_3 можна перейти до стану S_2 та S_7 , то можна розрахувати ймовірність переходу системи з стану S_3 до стану S_2 - P_{32} та ймовірність переходу системи з стану S_3 до стану S_7 - P_{37} :

- ймовірність переходу стану S_3 в S_2 :

$$\text{Оскільки } P_3 = \frac{1}{\mu+a\lambda}, P_{32} = \mu \cdot P_3 = \mu \cdot \frac{1}{\mu+a\lambda} = \frac{\mu}{\mu+a\lambda}.$$

- ймовірність переходу стану S_3 в S_7 :

$$P_{37} = a \cdot \lambda \cdot P_3 = a \cdot \lambda \cdot \frac{1}{\mu+a\lambda} = \frac{a\lambda}{\mu+a\lambda}.$$

5) для стану S_4 :

Оскільки зі стану S_4 можна перейти лише в стан S_1 , то ймовірність переходу стану S_4 в стан S_1 - $P_{41} = 1$.

б) для стану S_5 :

Оскільки зі стану S_5 можна перейти лише в стан S_9 , то ймовірність переходу стану S_5 в стан S_9 - $P_{59} = 1$.

7) для стану S_6 :

Оскільки з стану S_6 можна перейти до стану S_0 , S_7 , та стану S_8 , то можна розрахувати ймовірність переходу системи з стану S_6 до стану S_0 - P_{60} , ймовірність переходу системи з стану S_6 до стану S_7 - P_{67} та ймовірність переходу системи з стану S_6 до стану S_8 - P_{68} :

- ймовірність переходу стану S_6 в S_0 :

$$\text{Оскільки } P_6 = \frac{1}{\nu + \lambda}, \text{ то } P_{60} = \nu \cdot P_6 = \nu \cdot \frac{1}{\nu + \lambda} = \frac{\nu}{\nu + \lambda}.$$

- ймовірність переходу стану S_6 в S_7 :

$$P_{67} = a \cdot \lambda \cdot P_6 = a \cdot \lambda \cdot \frac{1}{\nu + \lambda} = \frac{a\lambda}{\nu + \lambda}.$$

- ймовірність переходу стану S_6 в S_8 :

$$P_{68} = (1-a) \cdot \lambda \cdot P_6 = (1-a) \cdot \lambda \cdot \frac{1}{\nu + \lambda} = \frac{(1-a)\lambda}{\nu + \lambda}.$$

8) для стану S_7 :

Оскільки зі стану S_7 можна перейти до стану S_1 та S_6 , то можна розрахувати ймовірність переходу системи з стану S_7 до стану S_1 - P_{71} та ймовірність переходу системи з стану S_7 до стану S_6 - P_{76} :

- ймовірність переходу стану S_7 в S_1 :

$$\text{Оскільки } P_7 = \frac{1}{\nu + \mu}, \text{ то } P_{71} = \nu \cdot P_7 = \nu \cdot \frac{1}{\nu + \mu} = \frac{\nu}{\nu + \mu}.$$

- ймовірність переходу стану S_7 в S_6 :

$$P_{76} = \mu \cdot P_7 = \mu \cdot \frac{1}{\nu + \mu} = \frac{\mu}{\nu + \mu}.$$

9) для стану S_8 :

Оскільки зі стану S_8 можна перейти до стану S_2 та S_9 , то можна розрахувати ймовірність переходу системи з стану S_8 до стану S_2 - P_{82} та ймовірність переходу системи з стану S_8 до стану S_9 - P_{89} :

- ймовірність переходу стану S_8 в S_2 :

Оскільки $P_8 = \frac{1}{\nu+a\lambda}$, то $P_{82} = \nu \cdot P_8 = \nu \cdot \frac{1}{\nu+a\lambda} = \frac{\nu}{\nu+a\lambda}$.

- ймовірність переходу стану S_8 в S_9 :

$$P_{89} = a \cdot \lambda \cdot P_8 = a \cdot \lambda \cdot \frac{1}{\nu+a\lambda} = \frac{a\lambda}{\nu+a\lambda}.$$

10) для стану S_9 :

Оскільки зі стану S_9 можна перейти лише в стан S_6 , то ймовірність переходу стану S_9 в стан S_6 - $P_{96} = 1$.

За допомогою отриманих ймовірностей можемо побудувати матрицю переходів:

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \left[\begin{array}{cccccccccc} 0 & a & (1-a) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\mu}{\mu+\lambda} & 0 & 0 & \frac{(1-a)\lambda}{\mu+\lambda} & \frac{a\lambda}{\mu+\lambda} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\mu}{\mu+a\lambda} & (1-a) & a & 0 & \frac{a\lambda}{\mu+a\lambda} & 0 & 0 \\ 0 & 0 & \frac{\mu}{\mu+a\lambda} & (1-a) & 0 & 0 & 0 & \frac{a\lambda}{\mu+a\lambda} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\nu}{\nu+\lambda} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(1-a)\lambda}{\nu+\lambda} & 0 \\ \frac{\nu}{\nu+\lambda} & \frac{\nu}{\nu+\mu} & 0 & 0 & 0 & 0 & \frac{\mu}{\nu+\mu} & \frac{\nu}{\nu+\lambda} & \frac{(1-a)\lambda}{\nu+\lambda} & 0 \\ 0 & 0 & \frac{\nu}{\nu+a\lambda} & 0 & 0 & 0 & \frac{\mu}{\nu+\mu} & \frac{\nu}{\nu+\lambda} & 0 & \frac{a\lambda}{\nu+a\lambda} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right] \end{matrix}$$

При $N = 8 \Rightarrow$ матриця $P^N = P^8$ - транзитивна (всі елементи позитивні)
 \Rightarrow існує стаціонарний режим ланцюга з дискретним часом \Rightarrow існує режим і з неперервним часом.

Сформуємо систему рівнянь Чепмена-Колмогорова до даної СМО:

$$P_i(t + \Delta t) = \sum_k P_k(t) P_{ki}(\Delta t); \quad i = \overline{0, 9}$$

$$\left\{ \begin{array}{l} P_0(t + \Delta t) = P_0(t)P_{00}(\Delta t) + P_1(t)P_{10}(\Delta t) + P_6(t)P_{60}(\Delta t) \\ P_1(t + \Delta t) = P_0(t)P_{01}(\Delta t) + P_1(t)P_{11}(\Delta t) + P_4(t)P_{41}(\Delta t) + P_7(t)P_{71}(\Delta t) \\ P_2(t + \Delta t) = P_0(t)P_{02}(\Delta t) + P_2(t)P_{22}(\Delta t) + P_3(t)P_{32}(\Delta t) + P_8(t)P_{82}(\Delta t) \\ P_3(t + \Delta t) = P_1(t)P_{13}(\Delta t) + P_2(t)P_{23}(\Delta t) + P_3(t)P_{33}(\Delta t) \\ P_4(t + \Delta t) = P_1(t)P_{14}(\Delta t) + P_4(t)P_{44}(\Delta t) \\ P_5(t + \Delta t) = P_2(t)P_{25}(\Delta t) + P_5(t)P_{55}(\Delta t) \\ P_6(t + \Delta t) = P_6(t)P_{66}(\Delta t) + P_7(t)P_{76}(\Delta t) + P_9(t)P_{96}(\Delta t) \\ P_7(t + \Delta t) = P_3(t)P_{37}(\Delta t) + P_6(t)P_{67}(\Delta t) + P_7(t)P_{77}(\Delta t) \\ P_8(t + \Delta t) = P_6(t)P_{68}(\Delta t) + P_8(t)P_{88}(\Delta t) \\ P_9(t + \Delta t) = P_5(t)P_{59}(\Delta t) + P_8(t)P_{89}(\Delta t) + P_9(t)P_{99}(\Delta t) \end{array} \right.$$

Підставляємо розраховані ймовірності в систему рівнянь:

$$\left\{ \begin{array}{l} P_0(t + \Delta t) = P_0(t)(1 - a\lambda\Delta t - \lambda\Delta t + a\lambda\Delta t) + P_1(t)\mu\Delta t + P_6(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) = P_0(t)a\lambda\Delta t + P_1(t)(1 - a\lambda\Delta t - \lambda\Delta t + a\lambda\Delta t) + P_4(t)\mu\Delta t + \\ \quad + P_7(t)\nu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) = P_0(t)(1 - a)\lambda\Delta t + P_2(t)(1 - a\lambda\Delta t - \lambda\Delta t + a\lambda\Delta t) + P_3(t)\mu\Delta t + \\ \quad + P_8(t)\nu\Delta t + o(\Delta t) \\ P_3(t + \Delta t) = P_1(t)(1 - a)\lambda\Delta t + P_2(t)a\lambda\Delta t + P_3(t)(1 - \mu\Delta t - a\lambda\Delta t) + o(\Delta t) \\ P_4(t + \Delta t) = P_1(t)a\lambda\Delta t + P_4(t)(1 - \mu\Delta t) + o(\Delta t) \\ P_5(t + \Delta t) = P_2(t)(1 - a)\lambda\Delta t + P_5(t)(1 - a\lambda\Delta t) + o(\Delta t) \\ P_6(t + \Delta t) = P_6(t)(1 - a\lambda\Delta t - \lambda\Delta t + a\lambda\Delta t) + P_7(t)\mu\Delta t + P_9(t)\nu\Delta t + o(\Delta t) \\ P_7(t + \Delta t) = P_3(t)a\lambda\Delta t + P_6(t)a\lambda\Delta t + P_7(t)(1 - \nu\Delta t - \mu\Delta t) + o(\Delta t) \\ P_8(t + \Delta t) = P_6(t)(1 - a)\lambda\Delta t + P_8(t)(1 - \nu\Delta t - a\lambda\Delta t) + o(\Delta t) \\ P_9(t + \Delta t) = P_5(t)a\lambda\Delta t + P_8(t)a\lambda\Delta t + P_9(t)(1 - \nu\Delta t) + o(\Delta t) \end{array} \right.$$

$$\left\{ \begin{array}{l} P_0(t + \Delta t) = P_0(t)(1 - \lambda\Delta t) + P_1(t)\mu\Delta t + P_6(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) = P_0(t)a\lambda\Delta t + P_1(t)(1 - \lambda\Delta t) + P_4(t)\mu\Delta t + P_7(t)\nu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) = P_0(t)(1 - a)\lambda\Delta t + P_2(t)(1 - \lambda\Delta t) + P_3(t)\mu\Delta t + P_8(t)\nu\Delta t \\ \quad + o(\Delta t) \\ P_3(t + \Delta t) = P_1(t)(1 - a)\lambda\Delta t + P_2(t)a\lambda\Delta t + P_3(t)(1 - \mu - a\lambda)\Delta t + o(\Delta t) \\ P_4(t + \Delta t) = P_1(t)a\lambda\Delta t + P_4(t)(1 - \mu\Delta t) + o(\Delta t) \\ P_5(t + \Delta t) = P_2(t)(1 - a)\lambda\Delta t + P_5(t)(1 - a\lambda\Delta t) + o(\Delta t) \\ P_6(t + \Delta t) = P_6(t)(1 - \lambda\Delta t) + P_7(t)\mu\Delta t + P_9(t)\nu\Delta t + o(\Delta t) \\ P_7(t + \Delta t) = P_3(t)a\lambda\Delta t + P_6(t)a\lambda\Delta t + P_7(t)(1 - \nu - \mu)\Delta t + o(\Delta t) \\ P_8(t + \Delta t) = P_6(t)(1 - a)\lambda\Delta t + P_8(t)(1 - \nu - a\lambda)\Delta t + o(\Delta t) \\ P_9(t + \Delta t) = P_5(t)a\lambda\Delta t + P_8(t)a\lambda\Delta t + P_9(t)(1 - \nu\Delta t) + o(\Delta t) \end{array} \right.$$

$$\left\{ \begin{array}{l} P_0(t + \Delta t) = P_0(t) - P_0(t)\lambda\Delta t + P_1(t)\mu\Delta t + P_6(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) = P_0(t)a\lambda\Delta t + P_1(t) - P_1(t)\lambda\Delta t + P_4(t)\mu\Delta t + P_7(t)\nu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) = P_0(t)(1 - a)\lambda\Delta t + P_2(t) - P_2(t)\lambda\Delta t + P_3(t)\mu\Delta t + P_8(t)\nu\Delta t \\ \quad + o(\Delta t) \\ P_3(t + \Delta t) = P_1(t)(1 - a)\lambda\Delta t + P_2(t)a\lambda\Delta t + P_3(t) - P_3(t)\mu\Delta t - P_3(t)a\lambda\Delta t \\ \quad + o(\Delta t) \\ P_4(t + \Delta t) = P_1(t)a\lambda\Delta t + P_4(t) - P_4(t)\mu\Delta t + o(\Delta t) \\ P_5(t + \Delta t) = P_2(t)(1 - a)\lambda\Delta t + P_5(t) - P_5(t)a\lambda\Delta t + o(\Delta t) \\ P_6(t + \Delta t) = P_6(t) - P_6(t)\lambda\Delta t + P_7(t)\mu\Delta t + P_9(t)\nu\Delta t + o(\Delta t) \\ P_7(t + \Delta t) = P_3(t)a\lambda\Delta t + P_6(t)a\lambda\Delta t + P_7(t) - P_7(t)\nu\Delta t - P_7(t)\mu\Delta t + o(\Delta t) \\ P_8(t + \Delta t) = P_6(t)(1 - a)\lambda\Delta t + P_8(t) - P_8(t)\nu\Delta t - P_8(t)a\lambda\Delta t + o(\Delta t) \\ P_9(t + \Delta t) = P_5(t)a\lambda\Delta t + P_8(t)a\lambda\Delta t + P_9(t) - P_9(t)\nu\Delta t + o(\Delta t) \end{array} \right.$$

В правій частині рівнянь залишимо елементи, які мають Δt , а в ліву частину перенесемо інші, які не мають Δt :

$$\left\{ \begin{array}{l} P_0(t + \Delta t) - P_0(t) = -P_0(t)\lambda\Delta t + P_1(t)\mu\Delta t + P_6(t)\nu\Delta t + o(\Delta t) \\ P_1(t + \Delta t) - P_1(t) = P_0(t)a\lambda\Delta t - P_1(t)\lambda\Delta t + P_4(t)\mu\Delta t + P_7(t)\nu\Delta t + o(\Delta t) \\ P_2(t + \Delta t) - P_2(t) = P_0(t)(1 - a)\lambda\Delta t - P_2(t)\lambda\Delta t + P_3(t)\mu\Delta t + P_8(t)\nu\Delta t \\ \quad + o(\Delta t) \\ P_3(t + \Delta t) - P_3(t) = P_1(t)(1 - a)\lambda\Delta t + P_2(t)a\lambda\Delta t - P_3(t)\mu\Delta t - P_3(t)a\lambda\Delta t \\ \quad + o(\Delta t) \\ P_4(t + \Delta t) - P_4(t) = P_1(t)a\lambda\Delta t - P_4(t)\mu\Delta t + o(\Delta t) \\ P_5(t + \Delta t) - P_5(t) = P_2(t)(1 - a)\lambda\Delta t - P_5(t)a\lambda\Delta t + o(\Delta t) \\ P_6(t + \Delta t) - P_6(t) = -P_6(t)\lambda\Delta t + P_7(t)\mu\Delta t + P_9(t)\nu\Delta t + o(\Delta t) \\ P_7(t + \Delta t) - P_7(t) = P_3(t)a\lambda\Delta t + P_6(t)a\lambda\Delta t - P_7(t)\nu\Delta t - P_7(t)\mu\Delta t + o(\Delta t) \\ P_8(t + \Delta t) - P_8(t) = P_6(t)(1 - a)\lambda\Delta t - P_8(t)\nu\Delta t - P_8(t)a\lambda\Delta t + o(\Delta t) \\ P_9(t + \Delta t) - P_9(t) = P_5(t)a\lambda\Delta t + P_8(t)a\lambda\Delta t - P_9(t)\nu\Delta t + o(\Delta t) \end{array} \right.$$

$$\left\{ \begin{array}{l}
\frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} = -P_0(t)\lambda + P_1(t)\mu + P_6(t)\nu \\
\frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = P_0(t)a\lambda - P_1(t)\lambda + P_4(t)\mu + P_7(t)\nu \\
\frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} = P_0(t)(1 - a)\lambda - P_2(t)\lambda + P_3(t)\mu + P_8(t)\nu \\
\frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} = P_1(t)(1 - a)\lambda + P_2(t)a\lambda - P_3(t)\mu - P_3(t)a\lambda \\
\frac{P_4(t + \Delta t) - P_4(t)}{\Delta t} = P_1(t)a\lambda - P_4(t)\mu \\
\frac{P_5(t + \Delta t) - P_5(t)}{\Delta t} = P_2(t)(1 - a)\lambda - P_5(t)a\lambda \\
\frac{P_6(t + \Delta t) - P_6(t)}{\Delta t} = -P_6(t)\lambda + P_7(t)\mu + P_9(t)\nu \\
\frac{P_7(t + \Delta t) - P_7(t)}{\Delta t} = P_3(t)a\lambda + P_6(t)a\lambda - P_7(t)\nu - P_7(t)\mu \\
\frac{P_8(t + \Delta t) - P_8(t)}{\Delta t} = P_6(t)(1 - a)\lambda - P_8(t)\nu - P_8(t)a\lambda \\
\frac{P_9(t + \Delta t) - P_9(t)}{\Delta t} = P_5(t)a\lambda + P_8(t)a\lambda - P_9(t)\nu
\end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} = -P_0(t)\lambda + P_1(t)\mu + P_6(t)\nu \\ \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = P_0(t)a\lambda - P_1(t)\lambda + P_4(t)\mu + P_7(t)\nu \\ \frac{P_2(t + \Delta t) - P_2(t)}{\Delta t} = P_0(t)(1 - a)\lambda - P_2(t)\lambda + P_3(t)\mu + P_8(t)\nu \\ \frac{P_3(t + \Delta t) - P_3(t)}{\Delta t} = P_1(t)(1 - a)\lambda + P_2(t)a\lambda - P_3(t)(\mu + a\lambda) \\ \frac{P_4(t + \Delta t) - P_4(t)}{\Delta t} = P_1(t)a\lambda - P_4(t)\mu \\ \frac{P_5(t + \Delta t) - P_5(t)}{\Delta t} = P_2(t)(1 - a)\lambda - P_5(t)a\lambda \\ \frac{P_6(t + \Delta t) - P_6(t)}{\Delta t} = -P_6(t)\lambda + P_7(t)\mu + P_9(t)\nu \\ \frac{P_7(t + \Delta t) - P_7(t)}{\Delta t} = P_3(t)a\lambda + P_6(t)a\lambda - P_7(t)(\nu + \mu) \\ \frac{P_8(t + \Delta t) - P_8(t)}{\Delta t} = P_6(t)(1 - a)\lambda - P_8(t)(\nu + a\lambda) \\ \frac{P_9(t + \Delta t) - P_9(t)}{\Delta t} = P_5(t)a\lambda + P_8(t)a\lambda - P_9(t)\nu \end{array} \right.$$

При $\Delta t \rightarrow 0$, границі правих частин рівнянь існують, тож вони існують і зліва, тоді це похідні функції $P_i(t)$, $i=\overline{0,9}$. Тоді запишемо систему диференціальних рівнянь:

$$\left\{ \begin{array}{l} P'_0 = -P_0(t)\lambda + P_1(t)\mu + P_6(t)\nu \\ P'_1 = P_0(t)a\lambda - P_1(t)\lambda + P_4(t)\mu + P_7(t)\nu \\ P'_2 = P_0(t)(1 - a)\lambda - P_2(t)\lambda + P_3(t)\mu + P_8(t)\nu \\ P'_3 = P_1(t)(1 - a)\lambda + P_2(t)a\lambda - P_3(t)(\mu + a\lambda) \\ P'_4 = P_1(t)a\lambda - P_4(t)\mu \\ P'_5 = P_2(t)(1 - a)\lambda - P_5(t)a\lambda \\ P'_6 = -P_6(t)\lambda + P_7(t)\mu + P_9(t)\nu \\ P'_7 = P_3(t)a\lambda + P_6(t)a\lambda - P_7(t)(\nu + \mu) \\ P'_8 = P_6(t)(1 - a)\lambda - P_8(t)(\nu + a\lambda) \\ P'_9 = P_5(t)a\lambda + P_8(t)a\lambda - P_9(t)\nu \end{array} \right.$$

Маємо систему десяти лінійних диференціальних рівнянь першого порядку з постійними коефіцієнтами. Задамо початкові умови, щоб вирішити систему рівнянь Колмогорова й знайти ймовірності станів:

$$P_0(0) = 1;$$

$$P_1(0) = P_2(0) = P_3(0) = P_4(0) = P_5(0) = P_6(0) = P_7(0) = P_8(0) = P_9(0) = 0.$$

Для вирішення приведеної системи рівнянь використаємо математичний пакет Matlab. За допомогою команди `syms` оголошуємо змінні (рис. 3.11), які використовуються в цій системі та використовуємо функцію `dsolve()` для вирішення диференціальних рівнянь (рис. 3.12).

```
>> %Оголошуємо змінні
>> %де a - а, l - лямбда, t - час, n - ню, m - мя
>> syms a l t n m;
>>
```

Рис. 3.11. Оголошення змінних

```
>> %Вирішуємо систему диференціальних рівнянь
>> dsolve('Dx0(t)=-x0(t)*l+x1(t)*m+x6(t)*n', 'Dx1(t)=x0(t)*a*l-x1(t)*l+x4(t)*m+x7(t)*n',
'Dx2(t)=x0(t)*(1-a)*l-x2(t)*l+x3(t)*m+x8(t)*n', 'Dx3(t)=x1(t)*(1-a)*l+x2(t)*a*l-x3(t)*(m+a*l)',
'Dx4(t)=x1(t)*a*l-x4(t)*m', 'Dx5(t)=x2(t)*(1-a)*l-x5(t)*a*l', 'Dx6(t)=-x6(t)*l+x7(t)*m+x9(t)*n',
'Dx7(t)=x3(t)*a*l+x6(t)*a*l-x7(t)*(n+m)', 'Dx8(t)=x6(t)*(1-a)*l-x8(t)*(n+a*l)',
'Dx9(t)=x5(t)*a*l+x8(t)*a*l-x9(t)*n', 'x0(0)=1', 'x1(0)=0', 'x2(0)=0', 'x3(0)=0',
'x4(0)=0', 'x5(0)=0', 'x6(0)=0', 'x7(0)=0', 'x8(0)=0', 'x9(0)=0')
```

Рис. 3.12. Вирішення диференціальних рівнянь

Рішення даної системи диференціальних рівнянь:

$$\begin{aligned}
 P_0(t) &= P_0(a, \lambda, t, \nu, \mu); \\
 P_1(t) &= P_1(a, \lambda, t, \nu, \mu); \\
 P_2(t) &= P_2(a, \lambda, t, \nu, \mu); \\
 P_3(t) &= P_3(a, \lambda, t, \nu, \mu); \\
 P_4(t) &= P_4(a, \lambda, t, \nu, \mu); \\
 P_5(t) &= P_5(a, \lambda, t, \nu, \mu); \\
 P_6(t) &= P_6(a, \lambda, t, \nu, \mu); \\
 P_7(t) &= P_7(a, \lambda, t, \nu, \mu); \\
 P_8(t) &= P_8(a, \lambda, t, \nu, \mu); \\
 P_9(t) &= P_9(a, \lambda, t, \nu, \mu).
 \end{aligned}$$

Знайдемо остаточні ймовірності:

$$\lim_{t \rightarrow \infty} P_i(t) = P_i; \quad i = \overline{0,9}; \quad \sum_{i=0}^9 P_i = 1.$$

Праві частини приведених рівнянь системи стають постійними, а, отже, й ліві теж постійні. Ці значення мають бути нулями, оскільки в іншому випадку $\lim_{t \rightarrow \infty} P_i(t)$ вийде з області допустимих значень $[0;1]$, що недопустимо. Тобто для знаходження ймовірностей прирівнюємо ліві частини системи рівнянь Колмогорова до нуля й вирішуємо тепер не диференціальну систему рівнянь, а лінійну.

$$\left\{ \begin{aligned}
 0 &= -P_0\lambda + P_1\mu + P_6\nu \\
 0 &= P_0a\lambda - P_1\lambda + P_4\mu + P_7\nu \\
 0 &= P_0(1-a)\lambda - P_2\lambda + P_3\mu + P_8\nu \\
 0 &= P_1(1-a)\lambda + P_2a\lambda - P_3(\mu + a\lambda) \\
 0 &= P_1a\lambda - P_4\mu \\
 0 &= P_2(1-a)\lambda - P_5a\lambda \\
 0 &= -P_6\lambda + P_7\mu + P_9\nu \\
 0 &= P_3a\lambda + P_6a\lambda - P_7(\nu + \mu) \\
 0 &= P_6(1-a)\lambda - P_8(\nu + a\lambda) \\
 0 &= P_5a\lambda + P_8a\lambda - P_9\nu
 \end{aligned} \right.$$

Для стаціонарного режиму система буде виглядами наступним чином:

$$\begin{cases} P_0\lambda = P_1\mu + P_6\nu \\ P_1\lambda = P_0a\lambda + P_4\mu + P_7\nu \\ P_2\lambda = P_0(1-a)\lambda + P_3\mu + P_8\nu \\ P_3(\mu + a\lambda) = P_1(1-a)\lambda + P_2a\lambda \\ P_4\mu = P_1a\lambda \\ P_5a\lambda = P_2(1-a)\lambda \\ P_6\lambda = P_7\mu + P_9\nu \\ P_7(\nu + \mu) = P_3a\lambda + P_6a\lambda \\ P_8(\nu + a\lambda) = P_6(1-a)\lambda \\ P_9\nu = P_5a\lambda + P_8a\lambda \end{cases}$$

Знову використаємо математичний пакет Matlab для знаходження розв'язку системи рівнянь. а допомогою команди `syms` оголошуємо змінні (рис. 3.13) та використовуємо функцію `solve()` для вирішення лінійних алгебраїчних рівнянь (рис. 3.14).

```
>> %Оголошуємо змінні
>> syms a l t n m;
>> syms x0 x1 x2 x3;
>>
```

Рис. 3.13. Оголошення змінних

```
>> %Вирішуємо лінійні алгебраїчні рівняння
>> solve('-x0*1+x1*m+x6*n=0', '-x1*1+x0*a*1+x4*m+x7*n=0', '-x2*1+x0*(1-a)*1+x3*m+x8*n=0',
'-x3*(m+a*1)+x1*(1-a)*1+x2*a*1=0', '-x4*m+x1*a*1=0', '-x5*a*1+x2*(1-a)*1=0',
'-x6*1+x7*m+x9*n=0', '-x7*(n+m)+x3*a*1+x6*a*1=0', '-x8*(n+a*1)+x6*(1-a)*1=0',
'-x9*n+x5*a*1+x8*a*1=0', 'x0+x1+x2+x3+x4+x5+x6+x7+x8+x9=1')
```

Рис. 3.14. Вирішення лінійних алгебраїчних рівнянь

$$\begin{aligned}
x_0 &= \\
& (a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m))/n - ((m^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (1^*(1^*x_0 - (D^*x_0)/C))/n/n - (1^*x_0 - (D^*x_0)/C)/n - (1^*x_0^*(a - 1) - (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m))/a + (m^*(m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/ \\
& (C^m))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)/(a^1))/n + (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m)) \\
& / (a^1) + (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)/(a^1) - (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m)) + 1 \\
x_1 &= \\
& (D^*x_0)/(C^m) \\
x_2 &= \\
& -(((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m))/(a^1) \\
x_3 &= \\
& -(((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)/(a^1) \\
x_4 &= \\
& (D^a a^{1*x_0})/(C^m^2) \\
x_5 &= \\
& (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m))*(a - 1)/(a^2 * 1) \\
x_6 &= \\
& (1^*x_0 - (D^*x_0)/C)/n \\
x_7 &= \\
& -(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m))/n \\
x_8 &= \\
& (1^*x_0^*(a - 1) - (((m + n)^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)^*(m + a^1))/(a^1 - (D^{1*x_0^*(a - 1)})/(C^m))/a + (m^*(m + n)^*(a^{1*x_0} - (D^{1*x_0} \\
& - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (a^{1^*(1^*x_0 - (D^*x_0)/C))/n)/(a^1))/n \\
x_9 &= \\
& ((m^*(a^{1*x_0} - (D^{1*x_0})/(C^m) + (D^a a^{1*x_0})/(C^m)))/n + (1^*(1^*x_0 - (D^*x_0)/C))/n)/n
\end{aligned}$$

Рис. 3.15. Розв'язки лінійних алгебраїчних рівнянь

Отримуємо рішення рівнянь у вигляді:

$$\begin{aligned}
P_0(t) &= P_0(a, \lambda, \nu, \mu); \\
P_1(t) &= P_1(a, \lambda, \nu, \mu); \\
P_2(t) &= P_2(a, \lambda, \nu, \mu); \\
P_3(t) &= P_3(a, \lambda, \nu, \mu); \\
P_4(t) &= P_4(a, \lambda, \nu, \mu); \\
P_5(t) &= P_5(a, \lambda, \nu, \mu); \\
P_6(t) &= P_6(a, \lambda, \nu, \mu); \\
P_7(t) &= P_7(a, \lambda, \nu, \mu); \\
P_8(t) &= P_8(a, \lambda, \nu, \mu); \\
P_9(t) &= P_9(a, \lambda, \nu, \mu).
\end{aligned}$$

Також можна аналітично вирішити цю систему рівнянь, виразивши кожне P_i , де $i=0,1,2,3,4,5,6,7,8,9$ через P_0 і постійні C та D :

$$\begin{aligned}
C &= a^2 \lambda^3 - a^2 \lambda \mu^2 - a^2 \lambda \nu \mu + a^2 \lambda \nu^2 + a \lambda^2 \mu + 2a \lambda^2 \nu + a \lambda \mu^2 + \\
&+ a \lambda \nu \mu - a \lambda \nu^2 - a \nu \mu^2 - a \nu^2 \mu + \lambda \nu \mu + \lambda \nu^2 + \nu \mu^2 + \nu^2 \mu, \\
(1.9)
\end{aligned}$$

$$\begin{aligned}
D &= a^4 \lambda^3 \nu + a^3 \lambda^4 + a^3 \lambda^3 \nu + a^3 \lambda^2 \mu - a^3 \lambda^2 \mu^2 - a^3 \lambda^2 \nu \mu - \\
&- a^4 \lambda^3 \nu^2 + a^2 \lambda^3 \mu + 2a^2 \lambda^3 \nu + a^2 \lambda^2 \mu^2 + a^2 \lambda^2 \nu \mu + 2a^2 \lambda^2 \nu^2 - \\
&- a^2 \lambda \nu^2 \mu^2 - a^2 \lambda \nu^2 \mu + a \lambda^2 \nu \mu + a \lambda \nu \mu^2 + a \lambda \nu^2 \mu. \\
(1.10)
\end{aligned}$$

$$\begin{cases}
P_0 \lambda = P_1 \mu + P_6 \nu \\
P_1 \lambda = P_0 a \lambda + P_4 \mu + P_7 \nu \\
P_2 \lambda = P_0 (1 - a) \lambda + P_3 \mu + P_8 \nu \\
P_3 (\mu + a \lambda) = P_1 (1 - a) \lambda + P_2 a \lambda \\
P_4 \mu = P_1 a \lambda \\
P_5 a \lambda = P_2 (1 - a) \lambda \\
P_6 \lambda = P_7 \mu + P_9 \nu \\
P_7 (\nu + \mu) = P_3 a \lambda + P_6 a \lambda \\
P_8 (\nu + a \lambda) = P_6 (1 - a) \lambda \\
P_9 \nu = P_5 a \lambda + P_8 a \lambda
\end{cases}$$

$$P_1 = \frac{D}{\mu C} P_0, \quad (1.11)$$

$$P_4 = \frac{a \lambda}{\mu} P_1 = \frac{a \lambda}{\mu} \cdot \frac{D}{\mu C} P_0 = \frac{a \lambda D}{\mu^2 C} P_0, \quad (1.12)$$

$$P_6 = \frac{P_0 \lambda - P_1 \mu}{\nu} = \frac{P_0 \lambda - \frac{D P_0 \mu}{\mu C}}{\nu} = \frac{P_0 \mu (\lambda C - D)}{\mu C \nu} = \frac{P_0 (\lambda C - D)}{C \nu} =$$

$$= \frac{P_0(\lambda C - D)}{\nu C} = \frac{\lambda C - D}{\nu C} P_0, \quad (1.13)$$

$$\begin{aligned} P_7 &= \frac{P_1\lambda - P_0a\lambda - P_4\mu}{\nu} = \frac{\frac{P_0D\lambda}{\mu C} - P_0a\lambda - \frac{P_0a\lambda D\mu}{\mu^2 C}}{\nu} = \\ &= \frac{\frac{P_0D\lambda\mu - P_0a\lambda\mu^2 C - P_0a\lambda D\mu}{\mu^2 C}}{\nu} = \frac{P_0D\lambda\mu - P_0a\lambda\mu^2 C - P_0a\lambda D\mu}{\nu\mu^2 C} = \\ &= \frac{P_0\mu(D\lambda - a\lambda\mu C - a\lambda D)}{\nu\mu^2 C} = \frac{P_0(D\lambda - a\lambda\mu C - a\lambda D)}{\nu\mu C} = \frac{D\lambda - a\lambda\mu C - a\lambda D}{\nu\mu C} P_0, \\ P_7 &= \frac{D\lambda - a\lambda\mu C - a\lambda D}{\nu\mu C} P_0, \quad (1.14) \end{aligned}$$

$$\begin{aligned} P_9 &= \frac{P_6\lambda - P_7\mu}{\nu} = \frac{\frac{P_0(\lambda C - D)\lambda}{\nu C} - \frac{P_0(D\lambda - a\lambda\mu C - a\lambda D)\mu}{\nu\mu C}}{\nu} = \\ &= \frac{\frac{P_0(\lambda C - D)\lambda\mu - P_0(D\lambda - a\lambda\mu C - a\lambda D)\mu}{\nu\mu C}}{\nu} = \\ &= \frac{P_0(\lambda C - D)\lambda\mu - P_0(D\lambda - a\lambda\mu C - a\lambda D)\mu}{\nu^2\mu C} = \\ &= \frac{P_0\mu((\lambda C - D)\lambda - D\lambda + a\lambda\mu C + a\lambda D)}{\nu^2\mu C} = \frac{P_0((\lambda C - D)\lambda - D\lambda + a\lambda\mu C + a\lambda D)}{\nu^2 C} \\ &= \frac{(\lambda C - D)\lambda - D\lambda + a\lambda\mu C + a\lambda D}{\nu^2 C} P_0 = \frac{\lambda^2 C - D\lambda - D\lambda + a\lambda\mu C + a\lambda D}{\nu^2 C} P_0 = \\ &= \frac{\lambda^2 C - 2D\lambda + a\lambda\mu C + a\lambda D}{\nu^2 C} P_0, \\ P_9 &= \frac{\lambda^2 C - 2D\lambda + a\lambda\mu C + a\lambda D}{\nu^2 C} P_0, \quad (1.15) \end{aligned}$$

$$\begin{aligned} P_3 &= \frac{P_7(\nu + \mu) - P_6a\lambda}{a\lambda} = \frac{\frac{P_0(D\lambda - a\lambda\mu C - a\lambda D)(\nu + \mu)}{\nu\mu C} - \frac{P_0(\lambda C - D)a\lambda}{\nu C}}{a\lambda} = \\ &= \frac{\frac{P_0(D\lambda - a\lambda\mu C - a\lambda D)(\nu + \mu) - P_0(\lambda C - D)a\lambda\mu}{\nu\mu C}}{a\lambda} = \\ &= \frac{P_0(D\lambda - a\lambda\mu C - a\lambda D)(\nu + \mu) - P_0(\lambda C - D)a\lambda\mu}{a\lambda\nu\mu C} = \end{aligned}$$

$$\begin{aligned}
&= \frac{P_0(D\lambda\nu - a\lambda\mu C\nu - a\lambda D\nu + D\lambda\mu - a\lambda\mu^2 C - a\lambda D\mu) - P_0(\lambda^2 C a\mu - D a\lambda\mu)}{a\lambda\nu\mu C} = \\
&= \frac{P_0(D\lambda\nu - a\lambda\mu C\nu - a\lambda D\nu + D\lambda\mu - a\lambda\mu^2 C - a\lambda D\mu - \lambda^2 C a\mu + D a\lambda\mu)}{a\lambda\nu\mu C} = \\
&= \frac{P_0\lambda(D\nu - a\mu C\nu - aD\nu + D\mu - a\mu^2 C - aD\mu - \lambda C a\mu + D a\mu)}{a\lambda\nu\mu C} = \\
&= \frac{(D\nu - a\mu C\nu - aD\nu + D\mu - a\mu^2 C - aD\mu - \lambda C a\mu + D a\mu)}{a\nu\mu C} P_0 = \\
&= \frac{(D\nu - a\mu C\nu - aD\nu + D\mu - a\mu^2 C - \lambda C a\mu)}{a\nu\mu C} P_0 = \\
&= P_0 \left(\frac{D}{a\mu C} - 1 - \frac{D}{\mu C} + \frac{D}{a\nu C} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} \right) = P_0 \left(\frac{D}{a\mu C} - 1 - \frac{D}{\mu C} + \frac{D}{a\nu C} - \frac{\mu - \lambda}{\nu} \right), \\
P_3 &= P_0 \left(\frac{D}{a\mu C} - 1 - \frac{D}{\mu C} + \frac{D}{a\nu C} - \frac{\mu - \lambda}{\nu} \right), \tag{1.16} \\
P_2 &= \frac{P_3(\mu + a\lambda) - P_1(1 - a)\lambda}{a\lambda} = \\
&= \frac{\frac{P_0(D\nu - a\mu C\nu - aD\nu + D\mu - a\mu^2 C - \lambda C a\mu)(\mu + a\lambda)}{a\nu\mu C} - \frac{P_0(1 - a)\lambda D}{\mu C}}{a\lambda} = \\
&= \frac{\frac{P_0(D\nu - a\mu C\nu - aD\nu + D\mu - a\mu^2 C - \lambda C a\mu)(\mu + a\lambda) - P_0(1 - a)\nu\lambda D}{a\nu\mu C}}{a\lambda} = \\
&= P_0 \left(\frac{D\nu\mu - a\mu^2 C\nu - aD\nu\mu + D\mu^2 - a\mu^3 C - \lambda C a\mu^2 + D\nu a\lambda - a^2\lambda\mu C\nu}{a^2\lambda\nu\mu C} + \right. \\
&\quad \left. + \frac{-a^2\lambda D\nu + D\mu a\lambda - a^2\lambda\mu^2 C - \lambda^2 C a^2\mu - a\nu\lambda D + a^2\nu\lambda D}{a^2\lambda\nu\mu C} \right) = \\
&= P_0 \left(\frac{\nu\mu D - a\nu\mu^2 C - a\nu\mu D + \mu^2 D - a\mu^3 C - a\lambda\mu^2 C - a^2\lambda\nu\mu C}{a^2\lambda\nu\mu C} + \right. \\
&\quad \left. + \frac{a\lambda\mu D - a^2\lambda\mu^2 C - a^2\lambda^2\mu C}{a^2\lambda\nu\mu C} \right) = P_0 \left(\frac{\nu\mu D}{a^2\lambda\nu\mu C} - \frac{a\nu\mu^2 C}{a^2\lambda\nu\mu C} - \frac{a\nu\mu D}{a^2\lambda\nu\mu C} + \frac{\mu^2 D}{a^2\lambda\nu\mu C} - \right. \\
&\quad \left. - \frac{a\mu^3 C}{a^2\lambda\nu\mu C} - \frac{a\lambda\mu^2 C}{a^2\lambda\nu\mu C} - \frac{a^2\lambda\nu\mu C}{a^2\lambda\nu\mu C} + \frac{a\lambda\mu D}{a^2\lambda\nu\mu C} - \frac{a^2\lambda\mu^2 C}{a^2\lambda\nu\mu C} - \frac{a^2\lambda^2\mu C}{a^2\lambda\nu\mu C} \right) = \\
&= P_0 \left(\frac{D}{a^2\lambda C} - \frac{\mu}{a\lambda} - \frac{D}{a\lambda C} + \frac{\mu D}{a^2\lambda\nu C} - \frac{\mu^2}{a\lambda\nu} - \frac{\mu}{a\nu} - 1 + \frac{D}{a\nu C} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} \right) =
\end{aligned}$$

$$\begin{aligned}
&= P_0 \left(\frac{D}{a^2 \lambda C} - \frac{D}{a \lambda C} + \frac{\mu D}{a^2 \lambda \nu C} + \frac{D}{a \nu C} \right) - \left(\frac{\mu}{a \lambda} + \frac{\mu^2}{a \lambda \nu} + \frac{\mu}{a \nu} + \frac{\mu}{\nu} + \frac{\lambda}{\nu} \right) - 1 = \\
&= P_0 \left(\frac{\nu D}{a^2 \lambda \nu C} - \frac{a \nu D}{a^2 \lambda \nu C} + \frac{\mu D}{a^2 \lambda \nu C} + \frac{a \lambda D}{a^2 \lambda \nu C} \right) - \left(\frac{\nu \mu}{a \lambda \nu} + \frac{\mu^2}{a \lambda \nu} + \frac{\lambda \mu}{a \lambda \nu} + \frac{a \lambda \mu}{a \lambda \nu} + \frac{a \lambda^2}{a \lambda \nu} \right) \\
&- 1) = P_0 \left(\frac{(\nu D - a \nu D + \mu D + a \lambda D)}{a^2 \lambda \nu C} \right) - \left(\frac{\nu \mu + \mu^2 + \lambda \mu + a \lambda \mu + a \lambda^2}{a \lambda \nu} \right) - 1 = \\
&= P_0 \left(\frac{D}{a^2 \lambda \nu C} (\nu - a \nu + \mu + a \lambda) - \left(\frac{\nu \mu + \mu^2 + \lambda \mu + a \lambda \mu + a \lambda^2}{a \lambda \nu} \right) - 1 \right), \\
P_2 &= P_0 \left(\frac{D}{a^2 \lambda \nu C} (\nu - a \nu + \mu + a \lambda) - \left(\frac{\nu \mu + \mu^2 + \lambda \mu + a \lambda \mu + a \lambda^2}{a \lambda \nu} \right) - 1 \right) \quad (1.17)
\end{aligned}$$

$$\begin{aligned}
P_8 &= \frac{P_2 \lambda - P_0 (1 - a) \lambda - P_3 \mu}{\nu} = \\
&\frac{P_0 (\nu \mu D - a \nu \mu^2 C - a \nu \mu D + \mu^2 D - a \mu^3 C - a \lambda \mu^2 C - a^2 \lambda \nu \mu C + a \lambda \mu D) \lambda}{a^2 \lambda \nu \mu C} \\
&= \frac{\quad}{\nu} + \\
&+ \frac{P_0 (-a^2 \lambda \mu^2 C - a^2 \lambda^2 \mu C) \lambda}{a^2 \lambda \nu \mu C} + \frac{-P_0 (1 - a) \lambda}{\nu} + \frac{-P_0 (D \nu - a \mu C \nu - a D \nu) \mu}{a \nu \mu C} + \\
&- \frac{P_0 (D \mu - a \mu^2 C - \lambda C a \mu) \mu}{a \nu \mu C} = \frac{P_0 (\lambda \nu \mu D - a \lambda \nu \mu^2 C - a \lambda \nu \mu D + \lambda \mu^2 D)}{a^2 \lambda \nu^2 \mu C} + \\
&+ \frac{P_0 (-a \lambda \mu^3 C - a \lambda^2 \mu^2 C - a^2 \lambda^2 \nu \mu C + a \lambda^2 \mu D - a^2 \lambda^2 \mu^2 C - a^2 \lambda^3 \mu C)}{a^2 \lambda \nu^2 \mu C} + \\
&+ \frac{P_0 (a \lambda - \lambda)}{\nu} + \frac{-P_0 (\nu \mu D - a \nu \mu^2 C - a \nu \mu D)}{a \nu^2 \mu C} = \frac{P_0 (\lambda \nu \mu D - a \lambda \nu \mu^2 C)}{a^2 \lambda \nu^2 \mu C} + \\
&+ \frac{P_0 (-a \lambda \nu \mu D + \lambda \mu^2 D - a \lambda \mu^3 C - a \lambda^2 \mu^2 C - a^2 \lambda^2 \nu \mu C + a \lambda^2 \mu D)}{a^2 \lambda \nu^2 \mu C} + \\
&+ \frac{P_0 (-a^2 \lambda^2 \mu^2 C - a^2 \lambda^3 \mu C)}{a^2 \lambda \nu^2 \mu C} + \frac{P_0 (a \lambda - \lambda) a^2 \lambda \nu \mu C}{a^2 \lambda \nu^2 \mu C} + \frac{P_0 (-\nu \mu D + a \nu \mu^2 C) a \lambda}{a^2 \lambda \nu^2 \mu C} + \\
&+ \frac{P_0 (a \nu \mu D) a \lambda}{a^2 \lambda \nu^2 \mu C} = P_0 \left(\frac{\lambda \nu \mu D}{a^2 \lambda \nu^2 \mu C} - \frac{a \lambda \nu \mu^2 C}{a^2 \lambda \nu^2 \mu C} - \frac{a \lambda \nu \mu D}{a^2 \lambda \nu^2 \mu C} + \frac{\lambda \mu^2 D}{a^2 \lambda \nu^2 \mu C} - \right. \\
&- \frac{a \lambda \mu^3 C}{a^2 \lambda \nu^2 \mu C} - \frac{a \lambda^2 \mu^2 C}{a^2 \lambda \nu^2 \mu C} - \frac{a^2 \lambda^2 \nu \mu C}{a^2 \lambda \nu^2 \mu C} + \frac{a \lambda^2 \mu D}{a^2 \lambda \nu^2 \mu C} - \frac{a^2 \lambda^2 \mu^2 C}{a^2 \lambda \nu^2 \mu C} - \frac{a^2 \lambda^3 \mu C}{a^2 \lambda \nu^2 \mu C} +
\end{aligned}$$

$$\begin{aligned}
& + \frac{a^3 \lambda^2 \nu \mu C}{a^2 \lambda \nu^2 \mu C} - \frac{a^2 \lambda^2 \nu \mu C}{a^2 \lambda \nu^2 \mu C} - \frac{a \lambda \nu \mu D}{a^2 \lambda \nu^2 \mu C} + \frac{a^2 \lambda \nu \mu^2 C}{a^2 \lambda \nu^2 \mu C} + \frac{a^2 \lambda \nu \mu D}{a^2 \lambda \nu^2 \mu C} \Big) = P_0 \left(\frac{D}{a^2 \nu C} - \right. \\
& - \frac{\mu}{a \nu} - \frac{D}{a \nu C} + \frac{\mu D}{a^2 \nu^2 C} - \frac{\mu^2}{a \nu^2} - \frac{\lambda \mu}{a \nu^2} - \frac{\lambda}{\nu} + \frac{\lambda D}{a \nu^2 C} - \frac{\lambda \mu}{\nu^2} - \frac{\lambda^2}{\nu^2} + \frac{a \lambda}{\nu} - \frac{\lambda}{\nu} - \frac{D}{a \nu C} + \\
& + \frac{\mu}{\nu} + \frac{D}{\nu C} \Big) = P_0 \left(\left(\frac{D}{a^2 \nu C} - \frac{D}{a \nu C} + \frac{\mu D}{a^2 \nu^2 C} + \frac{\lambda D}{a \nu^2 C} - \frac{D}{a \nu C} + \frac{D}{\nu C} \right) - \left(\frac{\mu}{a \nu} + \frac{\mu^2}{a \nu^2} + \right. \right. \\
& + \frac{\lambda \mu}{a \nu^2} + \frac{\lambda}{\nu} + \frac{\lambda \mu}{\nu^2} + \frac{\lambda^2}{\nu^2} - \frac{a \lambda}{\nu} + \frac{\lambda}{\nu} - \frac{\mu}{\nu} \Big) \Big) = P_0 \left(\left(\frac{\nu D}{a^2 \nu^2 C} - \frac{a \nu D}{a^2 \nu^2 C} + \frac{\mu D}{a^2 \nu^2 C} + \right. \right. \\
& + \frac{a \lambda D}{a^2 \nu^2 C} - \frac{a \nu D}{a^2 \nu^2 C} + \frac{a^2 \nu D}{a^2 \nu^2 C} \Big) - \left(\frac{\nu \mu}{a \nu^2} + \frac{\mu^2}{a \nu^2} + \frac{\lambda \mu}{a \nu^2} + \frac{a \lambda \nu}{a \nu^2} + \frac{a \lambda \mu}{a \nu^2} + \frac{\lambda a^2}{a \nu^2} - \frac{a^2 \lambda \nu}{a \nu^2} + \right. \\
& + \frac{a \lambda \nu}{a \nu^2} - \frac{a \nu \mu}{a \nu^2} \Big) \Big) = P_0 \left(\left(\frac{\nu D - a \nu D + \mu D + a \lambda D - a \nu D + a^2 \nu D}{a^2 \nu^2 C} \right) - \left(\frac{\nu \mu + \mu^2}{a \nu^2} + \right. \right. \\
& + \frac{\lambda \mu + a \lambda \nu + a \lambda \mu + a^2 \lambda - a^2 \lambda \nu + a \lambda \nu - a \nu \mu}{a \nu^2} \Big) \Big) = P_0 \left(\frac{D}{a^2 \nu^2 C} (\nu - a \nu + \mu + \right. \\
& + a \lambda - a \nu + a^2 \nu) - \frac{1}{a \nu^2} (\nu \mu + \mu^2 + \lambda \mu + a \lambda \nu + a \lambda \mu + a^2 \lambda - a^2 \lambda \nu + a \lambda \nu - \\
& - a \nu \mu) \Big) = P_0 \left(\frac{D}{a^2 \nu^2 C} (\nu - a \nu + \mu + \lambda - a \nu + a^2 \nu) - \frac{1}{a \nu^2} (\mu (\nu + \mu + \lambda + a \lambda - \right. \\
& - a \nu) + a \lambda (\nu + \lambda - a \nu + \nu)), \\
P_8 & = P_0 \left(\frac{D}{a^2 \nu^2 C} (\nu - a \nu + \mu + \lambda - a \nu + a^2 \nu) - \frac{1}{a \nu^2} (\mu (\nu + \mu + \lambda + a \lambda - \right. \\
& - a \nu) + a \lambda (\nu + \lambda - a \nu + \nu)), \tag{1.18}
\end{aligned}$$

$$\begin{aligned}
P_5 & = \frac{P_2 (1-a) \lambda}{a \lambda} = \frac{P_0 (D \nu \mu - a \mu^2 C \nu - a D \nu \mu + D \mu^2 - a \mu^3 C) (1-a) \lambda}{a^2 \lambda \nu \mu C} + \\
& + \frac{P_0 (-\lambda C a \mu^2 + D \nu a \lambda - a^2 \lambda \mu C \nu - a^2 \lambda D \nu + D \mu a \lambda - a^2 \lambda \mu^2 C) (1-a) \lambda}{a^2 \lambda \nu \mu C} + \\
& + \frac{P_0 (-\lambda^2 C a^2 \mu - a \nu \lambda D + a^2 \nu \lambda D) (1-a) \lambda}{a^2 \lambda \nu \mu C} = \frac{P_0 (D \nu \mu - a \mu^2 C \nu) (1-a) \lambda}{a^3 \lambda^2 \nu \mu C} + \\
& + \frac{P_0 (-a D \nu \mu + D \mu^2 - a \mu^3 C - \lambda C a \mu^2 + D \nu a \lambda - a^2 \lambda \mu C \nu - a^2 \lambda D \nu) (1-a) \lambda}{a^3 \lambda^2 \nu \mu C} + \\
& + \frac{P_0 (D \mu a \lambda - a^2 \lambda \mu^2 C - \lambda^2 C a^2 \mu - a \nu \lambda D + a^2 \nu \lambda D) (1-a) \lambda}{a^3 \lambda^2 \nu \mu C} =
\end{aligned}$$

$$\begin{aligned}
&= P_0 \left(\frac{\lambda\nu\mu D - a\lambda\nu\mu^2 C - a\lambda\nu\mu D + \lambda\mu^2 D - a\lambda\mu^3 C - a\lambda^2\mu^2 C + a\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \right. \\
&+ \frac{-a^2\lambda^2\nu\mu C - a^2\lambda^2\nu D + a\lambda^2\mu D - a^2\lambda^2\mu^2 C - a^2\lambda^3\mu C - a\lambda^2\mu D + a^2\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \\
&+ \frac{a\lambda\nu\mu D - a^2\lambda\nu\mu^2 C - a^2\lambda\nu\mu D + a\lambda\mu^2 D - a^2\lambda\mu^3 C - a^2\lambda^2\mu^2 C + a^2\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \\
&+ \left. \frac{-a^3\lambda^2\nu\mu C - a^3\lambda^2\nu D + a^2\lambda^2\mu D - a^3\lambda^2\mu^2 C - a^3\lambda^3\mu C - a^2\lambda^2\mu D + a^3\lambda^2\nu D}{a^3\lambda^2\nu\mu C} \right) \\
&= P_0 \left(\frac{\lambda\nu\mu D}{a^3\lambda^2\nu\mu C} - \frac{a\lambda\nu\mu^2 C}{a^3\lambda^2\nu\mu C} - \frac{a\lambda\nu\mu D}{a^3\lambda^2\nu\mu C} + \frac{\lambda\mu^2 D}{a^3\lambda^2\nu\mu C} - \frac{a\lambda\mu^3 C}{a^3\lambda^2\nu\mu C} - \frac{a\lambda^2\mu^2 C}{a^3\lambda^2\nu\mu C} + \right. \\
&+ \frac{a\lambda^2\nu D}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^2\nu\mu C}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \frac{a\lambda^2\mu D}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^2\mu^2 C}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^3\mu C}{a^3\lambda^2\nu\mu C} - \\
&- \frac{a\lambda^2\mu D}{a^3\lambda^2\nu\mu C} + \frac{a^2\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \frac{a\lambda\nu\mu D}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda\nu\mu^2 C}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda\nu\mu D}{a^3\lambda^2\nu\mu C} + \frac{a\lambda\mu^2 D}{a^3\lambda^2\nu\mu C} - \\
&- \frac{a^2\lambda\mu^3 C}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^2\mu^2 C}{a^3\lambda^2\nu\mu C} + \frac{a^2\lambda^2\nu D}{a^3\lambda^2\nu\mu C} - \frac{a^3\lambda^2\nu\mu C}{a^3\lambda^2\nu\mu C} - \frac{a^3\lambda^2\nu D}{a^3\lambda^2\nu\mu C} + \frac{a^2\lambda^2\mu D}{a^3\lambda^2\nu\mu C} - \\
&- \left. \frac{a^3\lambda^2\mu^2 C}{a^3\lambda^2\nu\mu C} - \frac{a^3\lambda^3\mu C}{a^3\lambda^2\nu\mu C} - \frac{a^2\lambda^2\mu D}{a^3\lambda^2\nu\mu C} + \frac{a^3\lambda^2\nu D}{a^3\lambda^2\nu\mu C} \right) = P_0 \left(\frac{D}{a^3\lambda C} - \frac{\mu}{a^2\lambda} - \right. \\
&- \frac{D}{a^2\lambda C} + \frac{\mu D}{a^3\lambda\nu C} - \frac{\mu^2}{a^2\lambda\nu} - \frac{\mu}{a^2\nu} + \frac{D}{a^2\mu C} - \frac{1}{a} - \frac{D}{a\mu C} + \frac{D}{a^2\nu C} - \frac{\mu}{a\nu} - \frac{\lambda}{a\nu} - \\
&- \frac{D}{a^2\nu C} + \frac{D}{a\mu C} + \frac{D}{a^2\lambda C} - \frac{\mu}{a\lambda} - \frac{D}{a\lambda C} + \frac{\mu D}{a^2\lambda\nu C} - \frac{\mu^2}{a\lambda\nu} - \frac{\mu}{a\nu} + \frac{D}{a\mu C} - 1 - \\
&- \frac{D}{\mu C} + \frac{D}{a\nu C} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} - \frac{D}{a\nu C} + \frac{D}{\mu C} \left. \right) = P_0 \left(\frac{D}{a^3\lambda C} - \frac{\mu}{a^2\lambda} + \frac{\mu D}{a^3\lambda\nu C} - \frac{\mu^2}{a^2\lambda\nu} - \frac{\mu}{a^2\nu} + \right. \\
&+ \frac{D}{a^2\mu C} - \frac{1}{a} - \frac{\mu}{a\nu} - \frac{\lambda}{a\nu} - \frac{\mu}{a\lambda} - \frac{D}{a\lambda C} + \frac{\mu D}{a^2\lambda\nu C} - \frac{\mu^2}{a\lambda\nu} - \frac{\mu}{a\nu} + \frac{D}{a\mu C} - 1 - \frac{\mu}{\nu} - \\
&- \left. \frac{\lambda}{\nu} \right) = P_0 \left(\frac{\mu D}{a^2\lambda\nu C} \left(\frac{1}{a} + 1 \right) + \frac{D}{a\lambda C} \left(\frac{1}{a^2} - 1 \right) + \frac{D}{a\mu C} \left(\frac{1}{a} + 1 \right) - \frac{\mu^2}{a\lambda\nu} \left(\frac{1}{a} + 1 \right) - \right. \\
&- \frac{\mu}{a\lambda} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{a\nu} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{\nu} \left(\frac{1}{a} + 1 \right) - \frac{\lambda}{\nu} \left(\frac{1}{a} + 1 \right) - \left. \left(\frac{1}{a} + 1 \right) \right) = \\
&= P_0 \left(\left(\frac{1}{a} + 1 \right) \left(\frac{\mu D}{a^2\lambda\nu C} + \frac{D}{a\mu C} - \frac{\mu^2}{a\lambda\nu} - \frac{\mu}{a\lambda} - \frac{\mu}{a\nu} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} - 1 \right) + \frac{D}{a\lambda C} \left(\frac{1}{a^2} - 1 \right) \right)
\end{aligned}$$

$$\begin{aligned}
&= P_0 \left(\left(\frac{1+a}{a} \right) \left(\frac{\mu D}{a^2 \lambda \nu C} + \frac{D}{a \mu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \lambda} - \frac{\mu}{a \nu} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} - 1 \right) + \frac{D}{a \lambda C} \left(\frac{1-a^2}{a^2} \right) \right) \\
P_5 &= P_0 \left(\left(\frac{1+a}{a} \right) \left(\frac{\mu D}{a^2 \lambda \nu C} + \frac{D}{a \mu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \lambda} - \frac{\mu}{a \nu} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} - 1 \right) + \right. \\
&\quad \left. + \frac{D}{a \lambda C} \left(\frac{1-a^2}{a^2} \right) \right) \tag{1.19}
\end{aligned}$$

Оскільки $P_0 + P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9 = 1$, то

$$\begin{aligned}
P_0 &= 1 - P_1 - P_2 - P_3 - P_4 - P_5 - P_6 - P_7 - P_8 - P_9 = 1 - \frac{D}{\mu C} P_0 - \\
&- P_0 \left(\frac{D}{a^2 \lambda C} - \frac{\mu}{a \lambda} - \frac{D}{a \lambda C} + \frac{\mu D}{a^2 \lambda \nu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \nu} - 1 + \frac{D}{a \nu C} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} \right) - \\
&- P_0 \left(\frac{D}{a \mu C} - 1 - \frac{D}{\mu C} + \frac{D}{a \nu C} - \frac{\mu - \lambda}{\nu} \right) - \frac{a \lambda D}{\mu^2 C} P_0 - P_0 \left(\frac{D}{a^3 \lambda C} - \frac{\mu}{a^2 \lambda} + \frac{\mu D}{a^3 \lambda \nu C} - \right. \\
&- \frac{\mu^2}{a^2 \lambda \nu} - \frac{\mu}{a^2 \nu} + \frac{D}{a^2 \mu C} - \frac{1}{a} - \frac{\mu}{a \nu} - \frac{\lambda}{a \nu} - \frac{\mu}{a \lambda} - \frac{D}{a \lambda C} + \frac{\mu D}{a^2 \lambda \nu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \nu} + \\
&+ \frac{D}{a \mu C} - 1 - \frac{\mu}{\nu} - \frac{\lambda}{\nu} \left. \right) - \frac{\lambda C - D}{\nu C} P_0 - \frac{D \lambda - a \lambda \mu C - a \lambda D}{\nu \mu C} P_0 - P_0 \left(\frac{D}{a^2 \nu C} - \right. \\
&- \frac{\mu}{a \nu} - \frac{D}{a \nu C} + \frac{\mu D}{a^2 \nu^2 C} - \frac{\mu^2}{a \nu^2} - \frac{\lambda \mu}{a \nu^2} - \frac{\lambda}{\nu} + \frac{\lambda D}{a \nu^2 C} - \frac{\lambda \mu}{\nu^2} - \frac{\lambda^2}{\nu^2} + \frac{a \lambda}{\nu} - \frac{\lambda}{\nu} - \frac{D}{a \nu C} + \\
&+ \frac{\mu}{\nu} + \frac{D}{\nu C} \left. \right) - \frac{\lambda^2 C - 2 D \lambda + a \lambda \mu C + a \lambda D}{\nu^2 C} P_0 = 1 - P_0 \left(\frac{D}{\mu C} + \frac{D}{a^2 \lambda C} - \frac{\mu}{a \lambda} - \right. \\
&- \frac{D}{a \lambda C} + \frac{\mu D}{a^2 \lambda \nu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \nu} - 1 + \frac{D}{a \nu C} - \frac{\mu}{\nu} - \frac{\lambda}{\nu} + \frac{D}{a \mu C} - 1 - \frac{D}{\mu C} + \frac{D}{a \nu C} - \\
&- \frac{\mu - \lambda}{\nu} + \frac{a \lambda D}{\mu^2 C} + \frac{D}{a^3 \lambda C} - \frac{\mu}{a^2 \lambda} + \frac{\mu D}{a^3 \lambda \nu C} - \frac{\mu^2}{a^2 \lambda \nu} - \frac{\mu}{a^2 \nu} + \frac{D}{a^2 \mu C} - \frac{1}{a} - \frac{\mu}{a \nu} - \frac{\lambda}{a \nu} - \\
&- \frac{\mu}{a \lambda} - \frac{D}{a \lambda C} + \frac{\mu D}{a^2 \lambda \nu C} - \frac{\mu^2}{a \lambda \nu} - \frac{\mu}{a \nu} + \frac{D}{a \mu C} - 1 - \frac{\mu}{\nu} - \frac{\lambda}{\nu} + \frac{\lambda C - D}{\nu C} + \\
&+ \frac{D \lambda - a \lambda \mu C - a \lambda D}{\nu \mu C} + \frac{D}{a^2 \nu C} - \frac{\mu}{a \nu} - \frac{D}{a \nu C} + \frac{\mu D}{a^2 \nu^2 C} - \frac{\mu^2}{a \nu^2} - \frac{\lambda \mu}{a \nu^2} - \frac{\lambda}{\nu} + \frac{\lambda D}{a \nu^2 C} - \\
&- \left. \frac{\lambda \mu}{\nu^2} - \frac{\lambda^2}{\nu^2} + \frac{a \lambda}{\nu} - \frac{\lambda}{\nu} - \frac{D}{a \nu C} + \frac{\mu}{\nu} + \frac{D}{\nu C} + \frac{\lambda^2 C - 2 D \lambda + a \lambda \mu C + a \lambda D}{\nu^2 C} \right) =
\end{aligned}$$

$$\begin{aligned}
&= 1 - P_0 \left(\frac{D}{a^2 \lambda C} + \frac{D}{a^2 \lambda C} + \frac{D}{a^3 \lambda C} + \frac{\mu D}{a^3 \lambda v C} + \frac{D}{a^2 \mu C} + \frac{\mu D}{a^2 \lambda v C} + \frac{D}{a^2 v C} + \frac{\mu D}{a^2 v^2 C} - \right. \\
&\quad - \frac{D}{a \lambda C} - \frac{D}{a \lambda C} + \frac{D}{a \mu C} + \frac{D}{a \mu C} + \frac{\lambda D}{v \mu C} - \frac{a \lambda D}{v \mu C} + \frac{\lambda D}{a v^2 C} - \frac{2 \lambda D}{v^2 C} + \frac{a \lambda D}{v^2 C} + \frac{a \lambda D}{\mu^2 C} - \\
&\quad - \frac{\mu}{a \lambda} - \frac{\mu^2}{a \lambda v} - \frac{\mu}{a v} - \frac{\mu}{v} - \frac{\mu}{a^2 \lambda} - \frac{\mu^2}{a^2 \lambda v} - \frac{\mu}{a^2 v} - \frac{\mu}{a v} - \frac{\mu}{a v} - \frac{\mu}{a \lambda v} - \frac{\mu}{a v} - \frac{\mu}{a v} - \frac{\mu}{v} - \\
&\quad - \frac{\mu^2}{a v^2} - \frac{\lambda \mu}{a v^2} - \frac{\lambda \mu}{v^2} + \frac{a \lambda \mu}{v^2} - \frac{\lambda}{v} - \frac{\lambda}{v} - \frac{\lambda}{v} - \frac{\lambda}{v} - \frac{\lambda}{a v} - 1 - 1 - 1 - \frac{1}{a} = \\
&= 1 - P_0 \left(\frac{\mu D}{a^2 \lambda v C} \left(\frac{1}{a} + 1 \right) + \frac{D}{a^2 \lambda C} \left(\frac{1}{a} + 1 \right) + \frac{D}{a \mu C} \left(\frac{1}{a} + 1 \right) + \frac{\lambda D}{v^2 C} \left(\frac{1}{a} + 1 \right) + \right. \\
&\quad + \frac{\lambda D}{v \mu C} (1 - a) + \frac{D}{v^2 C} \left(\frac{\mu}{a^2} - \lambda \right) - \frac{\lambda D}{C} \left(\frac{1}{v^2} + \frac{a}{\mu^2} \right) - \frac{D}{a \lambda C} \left(\frac{1}{a v} + 2 \right) + \frac{D}{a C} \left(\frac{1}{a v} + \frac{1}{\mu} \right) \\
&\quad - \frac{\mu}{a \lambda} \left(\frac{1}{a} + 1 \right) - \frac{\mu^2}{a \lambda v} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{a v} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{v} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{v} \left(\frac{1}{a} + 1 \right) - \\
&\quad - \frac{\lambda \mu}{v^2} \left(\frac{1}{a} + 1 \right) - \frac{\mu}{v^2} \left(\frac{\mu}{a} - a \lambda \right) - \frac{\mu}{a} \left(\frac{1}{\lambda} + \frac{1}{\lambda v} + \frac{1}{v} \right) - \frac{\lambda}{v} \left(\frac{1}{a} + 1 \right) - \frac{3 \lambda}{v} - \\
&\quad - 1 \left(\frac{1}{a} + 1 \right) - 2) = 1 - P_0 \left(\left(\frac{1}{a} + 1 \right) \left(\frac{\mu D}{a^2 \lambda v C} + \frac{D}{a^2 \lambda C} + \frac{D}{a \mu C} + \frac{\lambda D}{v^2 C} \right) + \right. \\
&\quad + \frac{\lambda D}{v \mu C} (1 - a) + \frac{D}{v^2 C} \left(\frac{\mu}{a^2} - \lambda \right) - \frac{\lambda D}{C} \left(\frac{1}{v^2} + \frac{a}{\mu^2} \right) - \frac{D}{a \lambda C} \left(\frac{1}{a v} + 2 \right) + \\
&\quad + \frac{D}{a C} \left(\frac{1}{a v} + \frac{1}{\mu} \right) - \left(\frac{1}{a} + 1 \right) \left(\frac{\mu}{a \lambda} + \frac{\mu^2}{a \lambda v} + \frac{\mu}{a v} + \frac{\mu}{v} + \frac{\mu}{v} + \frac{\lambda \mu}{v^2} \right) - \frac{\mu}{v^2} \left(\frac{\mu}{a} - a \lambda \right) - \\
&\quad - \frac{\mu}{a} \left(\frac{1}{\lambda} + \frac{1}{\lambda v} + \frac{1}{v} \right) - \frac{\lambda}{v} \left(\frac{1}{a} + 1 \right) - \frac{3 \lambda}{v} - 1 \left(\frac{1}{a} + 1 \right) - 2) = \\
&= 1 - P_0 \left(\left(\frac{1+a}{a} \right) \left(\frac{\mu D}{a^2 \lambda v C} + \frac{D}{a^2 \lambda C} + \frac{D}{a \mu C} + \frac{\lambda D}{v^2 C} \right) + \frac{\lambda D}{v \mu C} (1 - a) + \right. \\
&\quad + \frac{D}{v^2 C} \left(\frac{\mu - a^2 \lambda}{a^2} \right) - \frac{\lambda D}{C} \left(\frac{\mu^2 + a v^2}{v^2 \mu^2} \right) - \frac{D}{a \lambda C} \left(\frac{1 + 2 a v}{a v} \right) + \frac{D}{a C} \left(\frac{\mu + a v}{a v \mu} \right) - \\
&\quad - \left(\frac{1+a}{a} \right) \left(\frac{\mu}{a \lambda} + \frac{\mu^2}{a \lambda v} + \frac{\mu}{a v} + \frac{\mu}{v} + \frac{\mu}{v} + \frac{\lambda \mu}{v^2} \right) - \frac{\mu}{v^2} \left(\frac{\mu - a^2 \lambda}{a} \right) - \\
&\quad - \frac{\mu}{a} \left(\frac{1}{\lambda} + \frac{1}{\lambda v} + \frac{1}{v} \right) - \frac{\lambda}{v} \left(\frac{1+a}{a} \right) - \frac{3 \lambda}{v} - 1 \left(\frac{1+a}{a} \right) - 2) = \\
&= 1 - P_0 \left(\left(\frac{1+a}{a} \right) \left(\frac{\mu D}{a^2 \lambda v C} + \frac{D}{a^2 \lambda C} + \frac{D}{a \mu C} + \frac{\lambda D}{v^2 C} + \frac{\mu}{a \lambda} + \frac{\mu^2}{a \lambda v} + \frac{\mu}{a v} + \frac{2 \mu}{v} + \frac{\lambda \mu}{v^2} - \right. \right. \\
&\quad \left. \left. - \frac{\lambda}{v} - 1 \right) + \frac{\lambda D}{v \mu C} (1 - a) + \frac{D}{v^2 C} \left(\frac{\mu - a^2 \lambda}{a^2} \right) - \frac{\lambda D}{C} \left(\frac{\mu^2 + a v^2}{v^2 \mu^2} \right) - \frac{D}{a \lambda C} \left(\frac{1 + 2 a v}{a v} \right) + \right.
\end{aligned}$$

$$\begin{aligned}
& + \frac{D}{aC} \left(\frac{\mu + av}{av\mu} \right) - \frac{\mu}{v^2} \left(\frac{\mu - a^2\lambda}{a} \right) - \frac{\mu}{a} \left(\frac{1}{\lambda} + \frac{1}{\lambda v} + \frac{1}{v} \right) - \frac{3\lambda}{v} - 2). \\
P_0 = & \frac{1}{1 + \left(\frac{1+a}{a} \right) \left(\frac{\mu D}{a^2\lambda v C} + \frac{D}{a^2\lambda C} + \frac{D}{a\mu C} + \frac{\lambda D}{v^2 C} + \frac{\mu}{a\lambda} + \frac{\mu^2}{a\lambda v} + \frac{\mu}{av} + \frac{2\mu}{v} + \frac{\lambda\mu}{v^2} - \right.} \\
& \left. - \frac{\lambda}{v} - 1 \right) + \frac{\lambda D}{v\mu C} (1 - a) + \frac{D}{v^2 C} \left(\frac{\mu - a^2\lambda}{a^2} \right) - \frac{\lambda D}{C} \left(\frac{\mu^2 + av^2}{v^2\mu^2} \right) - \frac{D}{a\lambda C} \left(\frac{1 + 2av}{av} \right) +} \\
& \frac{1}{+ \frac{D}{aC} \left(\frac{\mu + av}{av\mu} \right) - \frac{\mu}{v^2} \left(\frac{\mu - a^2\lambda}{a} \right) - \frac{\mu}{a} \left(\frac{1}{\lambda} + \frac{1}{\lambda v} + \frac{1}{v} \right) - \frac{3\lambda}{v} - 2} \quad (1.20)
\end{aligned}$$

Розрахуємо показники ефективності даної двоканальної СМО:

- 1) відносна пропускна здатність:

$$Q = P_0 + P_1 + P_2 + P_6 \quad (1.21)$$

Запит може бути прийнятий на обслуговування системою, якщо вона повністю вільна або ж один з її каналів вільний.

- 2) абсолютна пропускна здатність:

$$A = a\lambda Q \quad (1.22)$$

Кількість обслужених коректних запитів за одиницю часу СМО.

- 3) ймовірність втрати коректного запита:

$$P_{vtr} = 1 - Q \quad (1.23)$$

- 4) інтенсивність потоку втрачених коректних запитів (кількість коректних запитів, які отримали відмову з одиницю часу)

$$N_{vtr} = a\lambda P_{vtr} = a\lambda(1 - Q) \quad (1.24)$$

3.3. Висновки до розділу

В третьому розділі кваліфікаційної роботи було реалізовано аналітичну модель систем обслуговування з шкідливими запитами для наступних моделей: одноканальної та двоканальної СМО, наведено всі можливі стани, в яких ці системи можуть перебувати, побудовано граф станів з інтенсивністю

переходів для кожної з систем, розраховано ймовірності переходу з одного стану в інший для всіх можливих станів СМО, на основі отриманих ймовірностей побудовано матриці переходів, на основі отриманих даних виведено формули для обчислень всіх можливих ймовірностей СМО.

При обчислення рівнянь застосовувались дві техніки обрахунків: перша - обчислення власноруч, друга – з використанням Matlab. В Matlab для вирішення диференціальних та лінійних алгебраїчних рівнянь було використано вбудовані функції – `dsolve` та `solve`, в роботі було приведено приклади використання цих команд до кожної з систем та результати, які були обчислені на кожному з кроків.

Представлені формули обчислення показників ефективності для кожної з систем обслуговування.

РОЗДІЛ 4

ІМІТАЦІЙНА МОДЕЛЬ СМО З ШКІДЛИВИМИ ЗАПИТАМИ

4.1. Опис середовища програмування

При написанні програми було прийнято рішення використовувати мову C++, яка відповідає вимогам щодо розроблення програмної моделі для СМО. Щодо мови C++, то це вдосконалена версія мови C, яка побудована на її фундаменті та синтаксі. При використанні мови C виникали різноманітні труднощі з написанням коду програми складної структури, це й стало основною причиною для створення нової мови програмування C++, в яку були додані можливості, які допомагають уникнути цих труднощів, зокрема це об'єктно-орієнтоване програмування (ООП), адже мова C не підтримувала ООП, також було додано розширений набір вбудованих функцій та доповнено стандартну бібліотеку. Мова програмування C++ була створена у 1979 році Б'ярном Страуструпом і отримала назву «C with Classes» (C з класами), але згодом у 1983 році цю назву було змінено на ту, яка актуальна й досі – C++ [15]. Більшість можливостей, які були доповнені до мови C, безумовно пов'язані з підтримкою технології ООП. Можна сказати, що мова програмування C++ стала об'єктно-орієнтованою версією мови програмування C, але в її основу ввійшла не лише мова C, а й засоби

Simula67, як стверджує Б'ярн Страуструп, тому мова програмування C++ є результатом поєднання двох технологій програмування [16].

Мова програмування C++ володіє багатьма перевагами:

- розробка програм для різноманітних платформ та систем;
- висока продуктивність та швидкодія;
- доступні різні стилі та технології програмування;
- вбудовані бібліотеки для виконання багатьох операцій;
- простота у вивченні та використанні;
- доступне використання, за рахунок наявності редакторів коду, які містять компоненти C++ для розробки програм.

Для написання коду було використано редактор коду Visual Code 2017.

4.2. Загальний принцип роботи програми

Кількість каналів, час моделювання системи та інтенсивності потоків задаються програмно. Далі відбувається генерація потоку запитів, які підпорядковуються розподілу Пуассона з заданим параметром, який надходить на канал обслуговування. Час обслуговування коректного запиту та час ремонту каналу виведеного з ладу підпорядковуються експоненціальному закону з заданими параметрами.

Коли на канал надходить запит, то канал перевіряється на працездатність, можливі наступні випадки:

- 1) якщо канал вільний, то запит надходить на цей канал
 - якщо цей запит був шкідливим, то канал виводиться з ладу, з потоку «відкидається» цей запит, починається ремонт каналу та відлік часу, протягом якого він буде перебувати в цьому стані;

- якщо запит був коректний, то канал починає обробляти цей запит, починається відлік часу обробки цього запиту та після обробки цей запит «відкидається» з потоку;
 - якщо запит не надійшов, то починається відлік часу, протягом якого канал буде очікувати на запит і знаходитися в так званому стані простою;
- 2) якщо канал зайнятий, то запит надходить на наступний канал, якщо він є, якщо ж ні, то запит втрачається, для цього каналу визначається який тип запиту зараз знаходиться в нього
- якщо запит коректний, то виконуються ті ж дії, які наведені в пункті 1 для відповідного типу запиту;
 - якщо запит шкідливий, то виконуються ті ж дії, які наведені в пункті 1 для відповідного типу запиту;
- 3) якщо канал виведений з ладу, то запит надходить на наступний канал, а для цього каналу починається рахуватися час перебування в стану виведеному з ладу, якщо канали виведені з ладу, то запит втрачається.

Також програма зберігає всі стани, в них перебувала система, та час знаходження в них та заносить ці дані до файлу. Також до файлу вносяться й наступні параметри, які під час моделювання програма фіксує: кількість оброблених та втрачених коректних запитів, продуктивність кожного каналу обслуговування (кількість оброблених запитів за одну одиницю часу) та продуктивність всієї системи.

4.3. Опис програмної частини

В ролі основних модулів програми імітації СМО є класи ServiceChannel та ServiceSystem.

Клас ServiceChannel відповідає за діяльність кожного каналу СМО. В даному класі перевіряється стан каналу та тип запиту, який до нього надійшов і за цими даними далі виконуються відповідні дії з параметрами,

самим каналом та запитом. В таблиці 1 наведені параметри, які він містить цей клас та їх початкові значення.

Таблиця 4.1

Параметри класу ServiceChannel

Параметр	Опис	Початкове значення
current_state	Поточний стан в якому перебуває канал. Можливі наступні стани: -1 – канал виведений з ладу	0

Закінчення таблиці 4.1

	0 – канал вільний 1 – канал зайнятий (обробляє запит)	
state_changed	Логічне значення, яке вказує, що стан каналу був змінений. Можливі значення: false – стан каналу не змінився true – стан каналу змінився	false
disabled	Логічне значення, яке вказує на те, що канал виведений з ладу. Можливі значення: false – канал в робочому стані true – канал виведений з ладу	false
processed_requests	Кількість оброблених каналом коректних запитів	0
number_of_requests	Загальна кількість запитів, які надійшли на канал обслуговування	0
processing_time	Загальний час, який затрачений при обробці коректних запитів	0

waiting_time	Загальний час, при якому канал був вільний	0
disabled_time	Загальний час, при якому канал був виведений з ладу	0

Клас ServiceSystem відповідає за діяльність всієї СМО. В даному класі генерується вхідний потік запитів, які подаються на канали обслуговування.

У файл «sequence.txt» виводиться стан потоку в кожен момент часу (рисунок 4.1), тобто скільки коректних та шкідливих заявок приходить в один і той же час кожного моменту часу.

```

sequence.txt - Notepad
File Edit Format View Help
correct: 0, malicious: 0
correct: 1, malicious: 0
correct: 0, malicious: 0
correct: 3, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 1
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 1, malicious: 0
correct: 0, malicious: 0
correct: 1, malicious: 1
correct: 0, malicious: 0
correct: 0, malicious: 0
correct: 0, malicious: 0

```

Рис. 4.1. Файл «sequence.txt»

У файл «process.txt» виводяться стани системи, скільки часу система перебувала в цьому стані та відсоткове відношення цього часу до загального часу системи (рисунок 4.2).

```

process.txt - Notepad
File Edit Format View Help
0 0 4 it = 0.002%
1 1 6 it = 0.003%
0 1 12 it = 0.006%
1 1 9 it = 0.0045%
0 1 4 it = 0.002%
1 1 2 it = 0.001%
1 0 24 it = 0.012%
1 1 8 it = 0.004%
1 0 3 it = 0.0015%
1 1 39 it = 0.0195%
0 1 10 it = 0.005%
1 1 8 it = 0.004%
1 0 1 it = 0.0005%
1 1 11 it = 0.0055%
1 0 2 it = 0.001%
1 1 27 it = 0.0135%
1 0 2 it = 0.001%
0 0 2 it = 0.001%
-1 0 3 it = 0.0015%
-1 1 14 it = 0.007%
0 1 8 it = 0.004%
1 1 12 it = 0.006%
1 0 9 it = 0.0045%
1 1 16 it = 0.008%
1 0 6 it = 0.003%
1 1 2 it = 0.001%
0 1 2 it = 0.001%
1 1 31 it = 0.0155%
0 1 3 it = 0.0015%

```

Рис. 4.2. Файл «process.txt»

У файлі «statistics.txt» відображена статистика роботи СМО, всі стани системи, загальний час знаходження системи в цих станах, відсоткове відношення загального часу знаходження системи у кожному з станів до загального часу системи, продуктивність кожного каналу, продуктивність системи, кількість обслужених та втрачених запитів.

```

statistics.txt - Notepad
File Edit Format View Help
##### STATISTICS #####
-1 -1 833 it = 0.4165%
-1 0 2592 it = 1.296%
-1 1 10109 it = 5.0545%
0 -1 1585 it = 0.7925%
0 0 9348 it = 4.674%
0 1 21142 it = 10.571%
1 -1 8308 it = 4.154%
1 0 27649 it = 13.8245%
1 1 118434 it = 59.217%

correct requests for channel:
channel 1: processed requests = 3827, 0.019135 for 1 it, number of correct requests = 23542
channel 2: processed requests = 3620, 0.0181 for 1 it, number of correct requests = 19938

TOTAL SYSTEM PERFORMANCE = 0.037235

TOTAL CORRECT REQUESTS WERE PROCESSED = 7447 = 29.7999%

TOTAL CORRECT REQUESTS WERE LOST = 17543 = 70.2001%

```

Рис. 4.3. Файл «statistics.txt»

В таблиці 4.2 наведені параметри, які містить клас ServiceSystem та їх початкові значення.

Таблиця 4.2

Параметри класу ServiceSystem

Параметр	Опис	Початкове значення
<code>vector<ServiceChannel> channels</code>	Вектор, який складається з множини класів ServiceChannel	Відсутнє, визначається в ході виконання програми згідно з заданою кількістю каналів обслуговування
<code>vector<vector<int>> state_history</code>	Вектор векторів, які зберігають всі стани системи у яких перебувала система та динаміку їх зміни	Відсутнє, визначається в ході виконання програми

map<vector<int>, int> state_statistics	Вектор всі можливих станів	Відсутнє, визначається в ході виконання програми
time	Час роботи системи	Задається програмно
number_of_lost_correct_requests	Кількість втрачених коректних запитів	0
number_of_processed_correct_reques ts	Кількість оброблених коректних запитів	0

Іноді трапляються випадки, коли доводиться розв'язувати однотипні задачі кілька разів, в таких випадках доцільним буде використання механічних обчислювальних процесів. Використовуючи їх, обраховують результати задачі в певній послідовності дій. Опис таких процесів називають алгоритмом. Алгоритмом також вважають формальне представлення, діючи по якому можна отримати розв'язок задачі.

Алгоритм – це чітке представлення порядку дій, які потрібно виконати, щоб розв'язати задачу [17]. За допомогою алгоритму описується процес отримання результатів за допомогою вхідних даних, адже для його отримання необхідно виконати такі дії:

- отримати та обробити вхідні дані;
- на основі вхідних даних обрахувати вихідні дані (результати);
- відобразити результати.

Розробка алгоритму включає в себе розбиття всього процесу отримання результату на етапи, які відповідають одній конкретній дії, результати яких можуть бути використані при вирішенні наступних етапів, всі ці етапи відображаються в правильному, послідовному напрямі. [18]. За допомогою

побудованого алгоритму вдається швидше зрозуміти суть програми та ознайомитися з усіма етапами її реалізації.

Алгоритм роботи імітаційної моделі СМО з шкідливими запитами з n каналами обслуговування представлений на рис. 4.4 – 4.6.

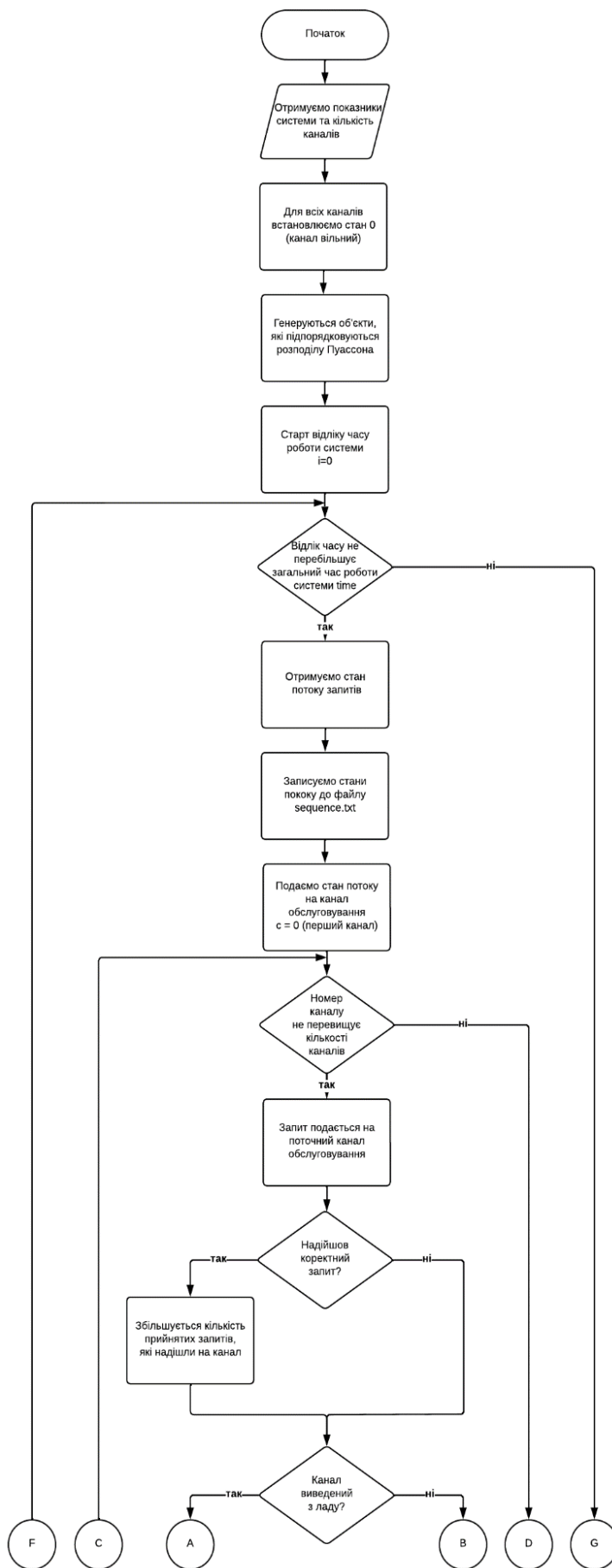


Рис. 4.4. Алгоритм роботи програми

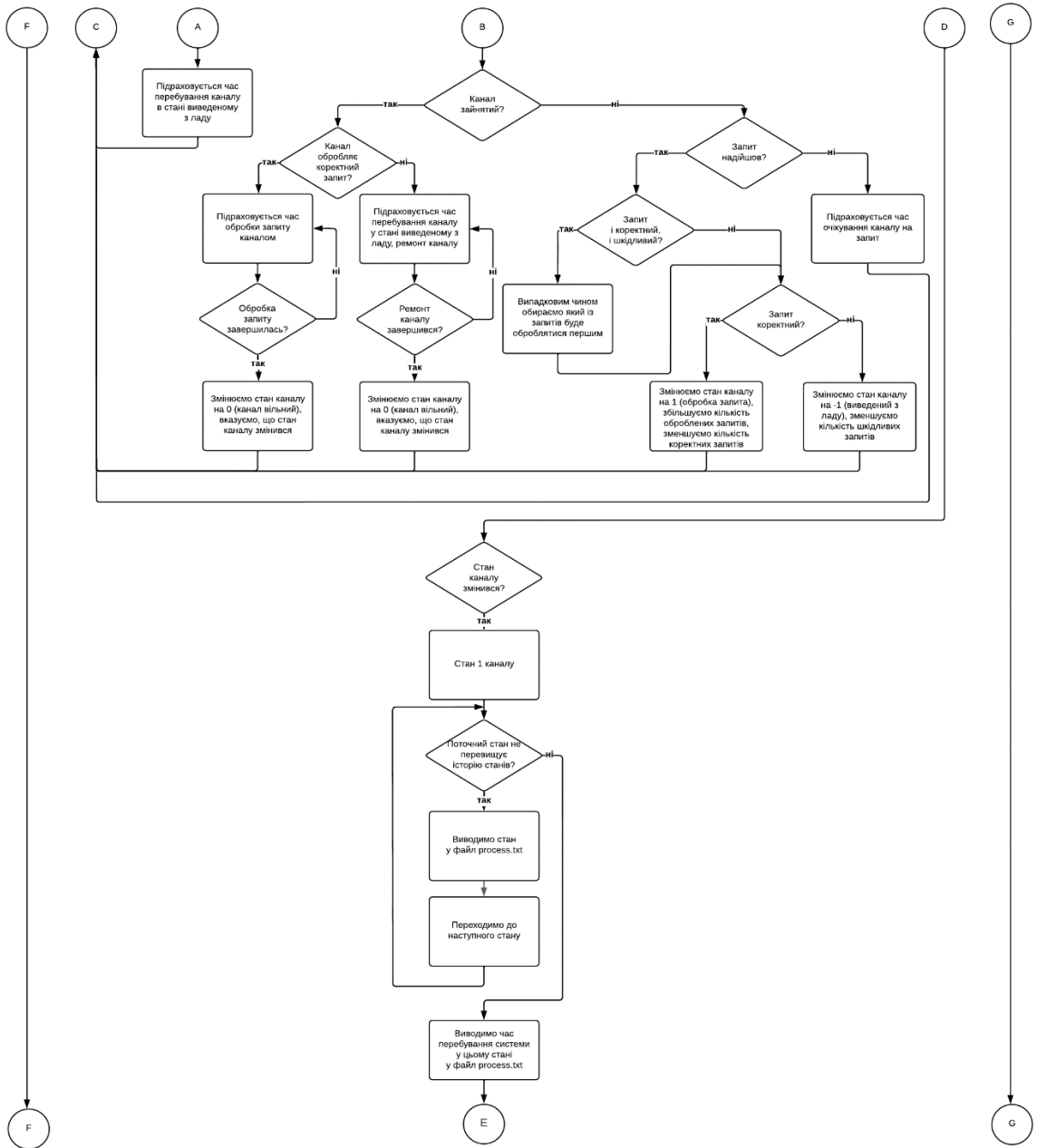


Рис. 4.5. Продовження алгоритму роботи програми

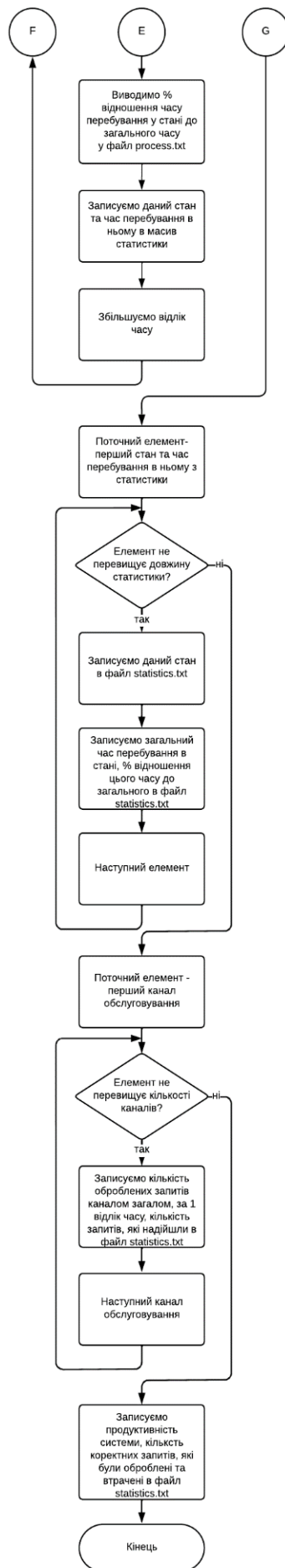


Рис. 4.6. Продовження алгоритму роботи програми

Діаграма класів [19] роботи СМО зображено на рис. 4.7.

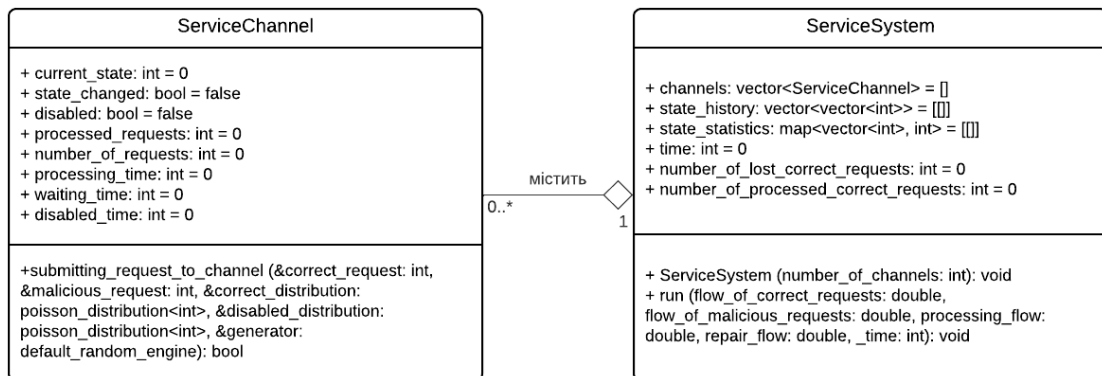


Рис. 4.7. Діаграма класів

Клас зображають у вигляді прямокутника, розділено на три секції: секція назви класу, секція атрибутів (значення цього класу) та секція операції з об'єктами іншого класу [20]. В розробленій програмі СМО міститься два класи – ServiceChannel та ServiceSystem.

Розглянемо кожен клас окремо:

- клас ServiceChannel

В першій секції вказано назву цього класу – ServiceChannel.

Друга секція відповідає за атрибути. Оскільки даний клас містить лише загальнодоступні значення (public), тому в цій секції перед назвою всіх значень додано символ +, адже він вказує на те, що атрибут public. Далі вказано перелік назв 8 атрибутів, які містить цей клас, їхній тип та початкове значення. Опис цих атрибутів був вже наведений в таблиці 4.1.

В цьому класі наявна лише одна операція, яка вказується у третій секції. Вона є загальнодоступною (public), тому перед її назвою вказаний символ +, по принципу, як і в другій секції. Далі в дужках вказані параметри, які приймає ця операція та вираз типу цих параметрів. Після дужок вказаний тип значення, який буде повернений цією операцією.

- клас ServiceSystem

В першій секції вказано назву цього класу – ServiceSystem.

Друга секція атрибутів. Оскільки даний клас містить лише загальнодоступні значення (public), тому в цій секції перед назвою всіх значень додано символ +. Далі наведено перелік назв 6 атрибутів, які містить цей клас, їхній тип та початкове значення. Опис цих атрибутів був вже наведений в таблиці 4.2.

Третя секція відповідає за операції. Даний клас містить 2 операції, вони є загальнодоступними, отже перед назвою операції знаходяться символи +, далі назви цих операцій, параметри, які вони приймають, вирази типу цих параметрів та тип значення, що повертають ці операції.

Дані класи зв'язані між собою бінарною асоціацією «містить» (порядок читання проходження класів – СМО містить клас) та мають відношення «частина-ціле» (відношення агрегації), адже агрегат – СМО містить у собі канал масового обслуговування (складова частина). Кожна із складових частин не наслідуює атрибути чи операції від цілого, а містять свої власні. Дана асоціація зображена суцільною лінією з незафарбованим ромбом, який вказує на ціле, інший клас – його складова частина.

4.4. Порівняння результатів аналітичної та імітаційної моделей

Для того, щоб оцінити коректність і надійність імітаційної моделі СМО необхідно порівняти отримані результати з результатами аналітичної моделі для одно та двоканальної систем при конкретно заданих параметрах для системи.

Задамо параметри для одноканальної СМО:

$$n = 1, a = 0,8, \lambda = 0,0425, \mu = 0,25, v = 0,05.$$

Програмне задання параметрів СМО:

```
int main() {
    srand(time(NULL));
    //задання показників системи і кількість каналів обслуговування
    double lambda = 0.0425, a = 0.8;
    double correct_flow = a * lambda, malicious_flow = (1 - a)*lambda;
    double processing_flow = 0.025, repair_flow = 0.05;
```

```

int time = 200000;
ServiceSystem system(1);
system.run(correct_flow, malicious_flow, processing_flow, repair_flow, time);
}

```

Підставляємо вказані параметри в формули (1.1) – (1.8):

$$P_0 = \frac{a\mu\nu}{a^2\lambda\nu + \mu\nu + a\lambda\mu - a^2\lambda\mu} = \frac{0,8 \cdot 0,25 \cdot 0,05}{0,8^2 \cdot 0,0425 \cdot 0,05 + 0,25 \cdot 0,05 + 0,8 \cdot 0,0425 \cdot 0,25 - 0,8^2 \cdot 0,0425 \cdot 0,25} =$$

$$= \frac{0,01}{0,00136 + 0,0125 + 0,0085 - 0,0068} = \frac{0,01}{0,01556} = 0,64267.$$

$$P_1 = \frac{a\lambda}{\mu} P_0 = \frac{0,8 \cdot 0,0425}{0,25} \cdot 0,64267 = \frac{0,034}{0,25} \cdot 0,64267 = 0,136 \cdot 0,64267 =$$

$$0,08740.$$

$$P_2 = \frac{1-a}{a} P_0 = \frac{1-0,8}{0,8} \cdot 0,64267 = \frac{1-0,8}{0,8} \cdot 0,64267 = 0,25 \cdot 0,64267 = 0,16067.$$

$$P_3 = \frac{(1-a)\lambda}{\nu} P_0 = \frac{(1-0,8) \cdot 0,0425}{0,05} \cdot 0,64267 = \frac{0,0085}{0,05} \cdot 0,64267 = 0,17 \cdot 0,64267 =$$

$$0,10925.$$

$$Q = P_0 = 0,64267.$$

$$A = a\lambda Q = 0,8 \cdot 0,0425 \cdot 0,64267 = 0,02185.$$

$$P_{vtr} = 1 - P_0 = 1 - 0,64267 = 0,35733.$$

$$B = a\lambda(1 - P_0) = 0,8 \cdot 0,0425 \cdot (1 - 0,64267) = 0,034 \cdot 0,35733 = 0,01215.$$

Заносимо дані розрахунків до таблиці 4.3, а також дані, які були отримано після моделювання імітаційної моделі системи з заданими параметрами, де

n – кількість каналів обслуговування,

A – абсолютна пропускна здатність,

P_{vtr} – ймовірність втрати коректного запита,

N_{vtr} – кількість втрачених коректних запитів за одиницю часу,

Q – відносна пропускна здатність.

Таблиця 4.3

Дані розрахунків аналітичної та імітаційної моделей для одноканальної СМО

n=1	Аналітична модель	Імітаційна модель	Різниця
A	0,02185	0,021938	-0,000088
P_{vtr}	0,35733	0,345383	0,011947
N_{vtr}	0,01215	0,011775	0,000375
Q	0,64267	0,64523	-0,00256

Задамо параметри для двоканальної СМО:

$$n = 2, a = 0,8, \lambda = 0,0425, \mu = 0,25, v = 0,05.$$

Програмне задання параметрів СМО:

```
int main() {
    srand(time(NULL));
    //задання показників системи і кількість каналів обслуговування
    double lambda = 0.0425, a = 0.8;
    double correct_flow = a * lambda, malicious_flow = (1 - a)*lambda;
    double processing_flow = 0.025, repair_flow = 0.05;
    int time = 20000;
    ServiceSystem system(2);
    system.run(correct_flow, malicious_flow, processing_flow, repair_flow, time);
}
```

Підставляємо вказані параметри в формули (1.9) – (1.24):

$$\begin{aligned}
 C = & 0,8^2 \cdot 0,0425^3 - 0,8^2 \cdot 0,0425 \cdot 0,25^2 - 0,8^2 \cdot 0,0425 \cdot 0,05 \cdot 0,25 + \\
 & + 0,8^2 \cdot 0,0425 \cdot 0,05^2 + 0,8 \cdot 0,0425^2 \cdot 0,25 + 2 \cdot 0,8 \cdot 0,0425^2 \cdot 0,05 + \\
 & + 0,8 \cdot 0,0425 \cdot 0,25^2 + 0,8 \cdot 0,0425 \cdot 0,05 \cdot 0,25 - 0,8 \cdot 0,0425 \cdot 0,05^2 - \\
 & - 0,8 \cdot 0,05 \cdot 0,25^2 - 0,8 \cdot 0,05^2 \cdot 0,25 + 0,0425 \cdot 0,05 \cdot 0,25 + \\
 & + 0,0425 \cdot 0,05^2 + 0,05 \cdot 0,25^2 + 0,05^2 \cdot 0,25 = 0,64 \cdot 0,00008 - 0,64 \cdot \\
 & \cdot 0,0425 \cdot 0,0625 - 0,64 \cdot 0,0425 \cdot 0,05 \cdot 0,25 + 0,64 \cdot 0,0425 \cdot 0,0025 + \\
 & + 0,8 \cdot 0,00181 \cdot 0,25 + 2 \cdot 0,8 \cdot 0,00181 \cdot 0,05 + 0,8 \cdot 0,0425 \cdot 0,0625 + \\
 & + 0,8 \cdot 0,0425 \cdot 0,05 \cdot 0,25 - 0,8 \cdot 0,0425 \cdot 0,0025 - 0,8 \cdot 0,05 \cdot 0,0625 - \\
 & - 0,8 \cdot 0,0025 \cdot 0,25 + 0,0425 \cdot 0,05 \cdot 0,25 + 0,0425 \cdot 0,0025 + \\
 & + 0,05 \cdot 0,0625 + 0,0025 \cdot 0,25 = 0,0000512 - 0,0017 - 0,00034 + \\
 & + 0,000068 + 0,000362 + 0,0001448 + 0,002125 + 0,000425 - 0,000085 - \\
 & - 0,0025 - 0,0005 + 0,00053 + 0,00011 + 0,003125 + 0,000625 = 0,00244.
 \end{aligned}$$

$$\begin{aligned}
D &= 0,8^4 \cdot 0,0425^3 \cdot 0,05 + 0,8^3 \cdot 0,0425^4 + 0,8^3 \cdot 0,0425^3 \cdot 0,05 + \\
&0,8^3 \cdot 0,0425^2 \cdot 0,25 - 0,8^3 \cdot 0,0425^2 \cdot 0,25^2 - 0,8^3 \cdot 0,0425^2 \cdot 0,05 \cdot 0,25 - \\
&- 0,8^4 \cdot 0,0425^3 \cdot 0,05^2 + 0,8^2 \cdot 0,0425^3 \cdot 0,25 + 2 \cdot 0,8^2 \cdot 0,0425^3 \cdot 0,05 + \\
&+ 0,8^2 \cdot 0,0425^2 \cdot 0,25^2 + 0,8^2 \cdot 0,0425^2 \cdot 0,05 \cdot 0,25 + 2 \cdot 0,8^2 \cdot 0,0425^2 \cdot \\
&\cdot 0,05^2 - 0,8^2 \cdot 0,0425 \cdot 0,05^2 \cdot 0,25^2 - 0,8^2 \cdot 0,0425 \cdot 0,05^2 \cdot 0,25 + \\
&+ 0,8 \cdot 0,0425^2 \cdot 0,05 \cdot 0,25 + 0,8 \cdot 0,0425 \cdot 0,05 \cdot 0,25^2 + 0,8 \cdot 0,0425 \cdot \\
&\cdot 0,05^2 \cdot 0,25 = 0,4096 \cdot 0,000077 \cdot 0,05 + 0,512 \cdot 0,000003 + \\
&+ 0,512 \cdot 0,000076 \cdot 0,05 + 0,512 \cdot 0,001806 \cdot 0,25 - \\
&- 0,512 \cdot 0,001806 \cdot 0,0625 - 0,512 \cdot 0,001806 \cdot 0,05 \cdot 0,25 - \\
&- 0,4096 \cdot 0,000077 \cdot 0,0025 + 0,64 \cdot 0,000077 \cdot 0,25 + \\
&+ 2 \cdot 0,64 \cdot 0,000077 \cdot 0,05 + 0,64 \cdot 0,001806 \cdot 0,0625 + \\
&+ 0,64 \cdot 0,001806 \cdot 0,05 \cdot 0,25 + 2 \cdot 0,64 \cdot 0,001806 \cdot 0,0025 - \\
&- 0,64 \cdot 0,0425 \cdot 0,0025 \cdot 0,0625 - 0,64 \cdot 0,0425 \cdot 0,0025 \cdot 0,25 + \\
&+ 0,8 \cdot 0,001806 \cdot 0,05 \cdot 0,25 + 0,8 \cdot 0,0425 \cdot 0,05 \cdot 0,0625 + \\
&+ 0,8 \cdot 0,0425 \cdot 0,0025 \cdot 0,25 = \\
&= 0,00000158 + 0,00000154 + 0,00000195 + 0,000231 - 0,000058 - \\
&- 0,0000116 - 0,00000008 + 0,0000123 + 0,0000049 + 0,000072 + \\
&+ 0,000014 + 0,0000058 - 0,0000042 - 0,000017 + 0,000018 + \\
&+ 0,000106 + 0,000021 = 0,0004.
\end{aligned}$$

$$\begin{aligned}
P_0 &= \frac{1}{1 + \left(\frac{1+0,8}{0,8}\right) \left(\frac{0,25 \cdot 0,0004}{0,8^2 \cdot 0,0425 \cdot 0,05 \cdot 0,00244} + \frac{0,0004}{0,8^2 \cdot 0,0425 \cdot 0,00244} + \right.} \\
&\frac{1}{\left. \frac{0,0004}{0,8 \cdot 0,25 \cdot 0,00244} + \frac{0,0425 \cdot 0,0004}{0,05^2 \cdot 0,00244} + \frac{0,25}{0,8 \cdot 0,0425} + \frac{0,25^2}{0,8 \cdot 0,0425 \cdot 0,05} \right.} \\
&\frac{1}{\left. + \frac{0,25}{0,8 \cdot 0,05} + \frac{2 \cdot 0,25}{0,05} + \frac{0,0425 \cdot 0,25}{0,05^2} - \frac{0,0425}{0,05} - 1 \right) +} \\
&\frac{1}{\left. + \frac{0,0425 \cdot 0,0004}{0,05 \cdot 0,25 \cdot 0,00244} (1 - 0,8) + \frac{0,0004}{0,05^2 \cdot 0,00244} \left(\frac{0,25 - 0,8^2 \cdot 0,0425}{0,8^2} \right) -} \right.}
\end{aligned}$$

$$\begin{aligned}
& - \frac{1}{\frac{0,0425 \cdot 0,0004}{0,00244} \left(\frac{0,25^2 + 0,8 \cdot 0,05^2}{0,05^2 \cdot 0,25^2} \right) - \frac{0,0004}{0,8 \cdot 0,0425 \cdot 0,00244}} \\
& \cdot \left(\frac{1 + 2 \cdot 0,8 \cdot 0,05}{0,8 \cdot 0,05} \right) + \frac{0,0004}{0,8 \cdot 0,00244} \left(\frac{0,25 + 0,8 \cdot 0,05}{0,8 \cdot 0,05 \cdot 0,25} \right) - \frac{0,25}{0,05^2} \\
& \cdot \left(\frac{0,25 - 0,8^2 \cdot 0,0425}{0,8} \right) - \frac{0,25}{0,8} \left(\frac{1}{0,0425} + \frac{1}{0,0425 \cdot 0,05} + \frac{1}{0,05} \right) - \\
& - \frac{1}{\frac{3 \cdot 0,0425}{0,05} - 2} = \frac{1}{1 + 2,25 \left(\frac{0,0001}{0,000003} + \frac{0,0004}{0,000066} + \frac{0,0004}{0,000488} + \frac{0,000017}{0,0000061} \right)} \\
& + \frac{0,25}{0,034} + \frac{0,0625}{0,0017} + \frac{0,25}{0,04} + \frac{0,5}{0,05} + \frac{0,010625}{0,0025} - \frac{0,0425}{0,05} - 1) + \\
& + \frac{0,0000034}{0,0000305} + \frac{0,0004}{0,0000061} \left(\frac{0,2228}{0,64} \right) - \frac{0,000017}{0,00244} \left(\frac{0,0645}{0,000156} \right) - \\
& - \frac{0,0004}{0,000082} \cdot \frac{1,08}{0,04} + \frac{0,0004}{0,001952} \cdot \frac{0,29}{0,01} - \frac{0,25}{0,0025} \cdot \frac{0,2228}{0,8} - \\
& - \frac{0,25}{0,8} \cdot 514,117 - \frac{0,1275}{0,05} - 2 = 0,193483, \\
P_1 &= \frac{0,0004}{0,25 \cdot 0,00244} \cdot 0,193483 = \frac{0,0004}{0,00061} \cdot 0,193483 = 0,65573 \cdot 0,193483 \\
&= 0,126873, \\
P_2 &= 0,193483 \cdot \left(\frac{0,0004}{0,8^2 \cdot 0,0425 \cdot 0,05 \cdot 0,00244} (0,05 - 0,8 \cdot 0,05 + 0,25 + \right. \\
& + 0,8 \cdot 0,0425) - \left. \frac{0,05 \cdot 0,25 + 0,25^2 + 0,0425 \cdot 0,25 + 0,8 \cdot 0,0425 \cdot 0,25 + \right. \\
& \left. + 0,8 \cdot 0,0425^2}{0,8 \cdot 0,0425 \cdot 0,05} \right) - 1) = 0,193483 \cdot 0,252176 = 0,048791, \\
P_3 &= 0,193483 \left(\frac{0,0004}{0,8 \cdot 0,25 \cdot 0,00244} - 1 + \frac{0,0004}{0,25 \cdot 0,00244} + \frac{0,0004}{0,8 \cdot 0,05 \cdot 0,00244} \right)
\end{aligned}$$

$$\begin{aligned}
& - \frac{0,25 - 0,0425}{0,05} \Big) = 0,193483 \left(\frac{0,0004}{0,000488} - 1 + \frac{0,0004}{0,00061} + \frac{0,0004}{0,0000976} - \right. \\
& \left. - \frac{0,2075}{0,05} \right) = 0,193483(0,81967 - 1 + 0,65573 + 4,09836 - 4,15) = \\
& = 0,193483 \cdot 0,42376 = 0,081990,
\end{aligned}$$

$$\begin{aligned}
P_4 &= \frac{0,8 \cdot 0,0425 \cdot 0,0004}{0,25^2 \cdot 0,00244} \cdot 0,193483 = \frac{0,000014}{0,0625 \cdot 0,00244} \cdot 0,193483 = \\
&= \frac{0,000014}{0,000152} \cdot 0,193483 = 0,09210 \cdot 0,193483 = 0,017819,
\end{aligned}$$

$$\begin{aligned}
P_5 &= 0,193483 \left(\left(\frac{1 + 0,8}{0,8} \right) \left(\frac{0,25 \cdot 0,0004}{0,8^2 \cdot 0,0425 \cdot 0,05 \cdot 0,00244} + \frac{0,0004}{0,8 \cdot 0,25 \cdot 0,00244} \right. \right. \\
& \left. \left. - \frac{0,25^2}{0,8 \cdot 0,0425 \cdot 0,05} - \frac{0,25}{0,8 \cdot 0,0425} - \frac{0,25}{0,8 \cdot 0,05} - \frac{0,25}{0,05} - \frac{0,0425}{0,05} - 1 \right) + \right. \\
& \left. + \frac{0,0004}{0,8 \cdot 0,0425 \cdot 0,00244} \left(\frac{1 - 0,8^2}{0,8^2} \right) \right) = 0,193483 \left(2,25 \cdot \left(\frac{0,0001}{0,0000033} + \right. \right. \\
& \left. \left. + \frac{0,0004}{0,000488} - \frac{0,0625}{0,0017} - \frac{0,25}{0,034} - \frac{0,25}{0,04} - 6,25 - 1,85 \right) + \frac{0,0004}{0,0000829} \cdot 0,5625 \right) = \\
&= 0,193483(2,7141 - 2,25 \cdot 1,135) = 0,193483(2,7141 - 2,55375) = \\
&= 0,193483 \cdot 0,16035 = 0,031025,
\end{aligned}$$

$$\begin{aligned}
P_6 &= \frac{0,0425 \cdot 0,00244 - 0,00004}{0,05 \cdot 0,00244} \cdot 0,193483 = \frac{0,000064}{0,000122} \cdot 0,193483 = \\
&= 0,52459 \cdot 0,193483 = 0,101499,
\end{aligned}$$

$$\begin{aligned}
P_7 &= \frac{0,0004 \cdot 0,0425 + 0,8 \cdot 0,0425 \cdot 0,25 \cdot 0,00244 -}{0,05 \cdot 0,25 \cdot 0,00244} \\
& - \frac{0,8 \cdot 0,00244 \cdot 0,0004}{0,05 \cdot 0,25 \cdot 0,00244} \cdot 0,193483 = \frac{0,000017 + 0,0000207 - 0,0000007}{0,0000305} \cdot \\
& \cdot 0,193483 = \frac{0,000037}{0,0000305} \cdot 0,193483 = 1,21311 \cdot 0,193483 = 0,234716,
\end{aligned}$$

$$\begin{aligned}
P_8 &= 0,193483 \left(\frac{0,0004}{0,8^2 \cdot 0,05^2 \cdot 0,00244} (0,05 - 0,8 \cdot 0,05 + 0,25 + 0,0425 - \right. \\
& \left. - 0,8 \cdot 0,05 + 0,8^2 \cdot 0,05) - \frac{1}{0,8 \cdot 0,5^2} (0,25(0,05 + 0,25 + 0,0425 + \right. \\
& \left. 0,8 \cdot 0,0425 - 0,8 \cdot 0,05) - 0,8 \cdot 0,0425(0,05 + 0,0425 - 0,8 \cdot 0,05 + 0,05) \right) =
\end{aligned}$$

$$\begin{aligned}
&= 0,193483 \left(\frac{0,0004}{0,00039} \cdot 0,2945 - 50(0,25 \cdot 0,3365 - 0,034 \cdot 0,1025) \right) = \\
&= 0,193483(1,02564 \cdot 0,2945 - 50(0,00841 - 0,0035)) = \\
&= 0,193483(0,302051 - 0,2455) = 0,053551 \cdot 0,193483 = 0,010941, \\
P_9 &= \frac{0,0425^2 \cdot 0,00244 + 2 \cdot 0,0004 \cdot 0,0425 - 0,8 \cdot 0,0425 \cdot 0,25 \cdot 0,00244}{0,05^2 \cdot 0,00244} - \\
&- \frac{0,8 \cdot 0,0425 \cdot 0,004}{0,05^2 \cdot 0,00244} \cdot 0,193483 = \frac{0,001806 \cdot 0,00244 + 0,000034 -}{0,0000061} \\
&- \frac{0,000020 - 0,0000136}{0,0000061} \cdot 0,193483 = \frac{0,00000481}{0,0000061} \cdot 0,193483 = \\
&= 0,788524 \cdot 0,193483 = 0,152566, \\
Q &= P_0 + P_1 + P_2 + P_6 = 0,193483 + 0,126873 + 0,048791 + 0,101499 = \\
&= 0,470646, \\
A &= a\lambda Q = 0,8 \cdot 0,0425 \cdot 0,470646 = 0,016001, \\
P_{vtr} &= 1 - Q = 1 - 0,470646 = 0,529354, \\
N_{vtr} &= a\lambda(1 - Q) = 0,8 \cdot 0,0425 \cdot (1 - 0,470646) = 0,8 \cdot 0,0425 \cdot 0,529354 \\
&= 0,017998.
\end{aligned}$$

Таблиця 4.4

Дані розрахунків аналітичної та імітаційної моделей для двоканальної СМО

n=2	Аналітична модель	Імітаційна модель	Різниця
A	0,016001	0,01601281	-0,00001181
P_{vtr}	0,529354	0,529035	0,000319
N_{vtr}	0,017998	0,020175	-0,002177
Q	0,470646	0,470965	-0,000319

Проаналізувавши таблицю 4.3 та таблицю 4.4, можна сказати, що погрішність обчислень відповідає тисячній долі і це є гарним показником. На основі цих таблиць можна зробити висновок, що імітаційна модель відповідає аналітичній, а значить є коректною та надійною.

4.5. Висновки до розділу

В четвертому розділі кваліфікаційної роботи було обрано мову для розробки імітаційної моделі - C++, оскільки вона не тільки відповідає всім виставленим вимогам щодо розроблення СМО, але й володіє багатьма перевагами, такими як:

- розробка програм для різноманітних платформ та систем;
- висока продуктивність та швидкодія;
- доступні різні стилі та технології програмування;
- вбудовані бібліотеки для виконання багатьох операцій;
- простота у вивченні та використанні;
- доступне використання, за рахунок наявності редакторів коду, які містять компоненти C++ для розробки програм.

Розробка імітаційної моделі відбувалася в редакторі коду Visual Code.

Описано загальну суть роботи розробленої імітаційної моделі та основні модулі програми - класи `ServiceChannel` та `ServiceSystem`. Представлено всі результати виконання програми з їх детальним описом.

Розроблено алгоритм роботи імітаційної моделі СМО з шкідливими запитами з n каналами обслуговування, який відображає чітку структуру виконання програми, діаграму класів, яка містить атрибути та операції класів `ServiceChannel` та `ServiceSystem` розробленої програми.

Проведено порівняльний аналіз результатів аналітичної та імітаційної моделей для СМО з одним та двома каналами обслуговування з деякими початковими даними та в результаті отримали цілком відповідну імітаційна модель аналітичній, що говорить про те, що розроблена імітаційна модель коректно працює.

ВИСНОВКИ

В ході написання кваліфікаційної роботи було виконано основне завдання, а саме: розробка імітаційної моделі системи обслуговування з шкідливими запитами.

В процесі розробки було досліджено систему масового обслуговування з шкідливими запитами з можливим ремонтом каналів, якщо він був виведений з ладу. Детально опрацьовано та проаналізовано шкідливі запити, вивчено їх сфери діяльності, різноманітні типи та дії. Розглянуто Реалізовано аналітичну модель систем обслуговування з шкідливими запитами для наступних моделей: одноканальної та двоканальної СМО, наведено всі можливі стани, в яких ці системи можуть перебувати, обчислені абсолютні та стаціонарні ймовірності, середнє значення кількості обслужених та втрачених коректних запитів за одиницю часу.

Було обрано об'єктно-орієнтовану мову програмування C++, як програмний засіб для розробки системи. Ця мова характеризується надійністю, зручністю, безпекою.

Для кращого розуміння системи, яка була розроблена, в роботі приведено загальних опис її діяльності, детально описано складові програмної частини, побудовано алгоритм роботи програми, діаграму класів, наведено та поетапно представлено отримані результати імітаційної моделі.

Проведений порівняльний аналіз між аналітичної та імітаційною моделями і виявлено, що вони цілком і повністю відповідають одна одній, тобто дану імітаційну модель можна практично застосовувати до реальних задач.

Результатом кваліфікаційної роботи є написана програма, яка дозволяє моделювати систему масового обслуговування з шкідливими запитами з n каналом(-ами) обслуговування. Тобто реалізована імітаційна модель, за

допомогою якої можна відстежувати зміни станів системи та отримувати показники ефективності цієї системи.

Таким чином, в кваліфікаційній роботі було досліджено новий тип задач теорії масового обслуговування, який раніше не вивчався в даній області. Можливе подальше вивчення та опрацювання даної тематики для застосування даної системи до задач, де шкідливі запити будуть розповсюджуються на інші канали чи обладнання.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Імітаційне моделювання систем масового обслуговування: навч. посіб. [для студентів техн. спец. вищ. навч. закл.] / В. Б. Толубко, А.Д. Кожухівський, В.В. Вишнівський, Г.І. Гайдур, О.А. Кожухівська. – Київ: 175 с.
2. Томашевський В. М. Моделювання систем. - К: Видавнича група ВНУ, 2005. 352 С.: іл. ISBN 966-552-120-9.
3. Задачин В. М. Моделювання систем : конспект лекцій / В. М. Задачин, І. Г. Конюшенко. – Харків : Вид. ХНЕУ, 2010. – 268 с.
4. Математичне та імітаційне моделювання СМО із застосуванням середовища Matlab. – К.: КНУБА, 2014. – 196 с.
5. Стеценко, І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2010. – 399 с.
6. Советов Б. Я., Яковлев С.А. Моделирование систем: Учеб. для вузов — 3-е юд., перераб. и доп. — М.: Высш. шк., 2001. — 343 с: ил.
7. Жерновий Ю. В. Імітаційне моделювання систем масового обслуговування: Практикум. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 307 с.
8. Великодний С.С. Моделювання систем: конспект лекцій. Одеський державний екологічний університет, 2018. – 186 с.
9. Дубчак Л.О., Бойків Н.І., Васильків Н.М. Система масового обслуговування книжкового магазину / Л.О. Дубчак, Н.І. Бойків, Н.М. Васильків// Інформатика, обчислювальна техніка та автоматизація. – Тернопіль, 2018. – Вип. 116. – С. 6. – Бібліогр.: 10 назв.
10. Моделі обслуговування – К.: НМетАУ, 2016. – 15 с.
11. Доля П.Г. Использование MATLAB. Решение дифференциальных уравнений./ П.Г. Доля// Харьковский Национальный Университет механико – математический факультет кафедра геометрии им. А.В. Погорелова. – Харків,

2008. – 52с.

12. Мурлін А.Г. Комп'ютерне моделювання: конспект лекцій. Кубанський державний технологічний університет, 2007. – 7 с.
13. Дьяконов В. П. MATLAB. Полный самоучитель. – М.: ДМК Пресс, 2012. – 768 с.: ил.
14. Дьоміна В. М. Оптимізаційні методи та моделі. Моделювання систем масового обслуговування: конспект лекцій / В. М. Дьоміна; Харк. нац. аграр. ун-т ім. В. В. Докучаєва. – Х.: ХНАУ, 2015. – 42 с.
15. Грицюк Ю.І., Рак Т.Є. Програмування мовою С++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с. – Статистика: іл. 10, табл. 18, бібліогр. 31.
16. Татарчук Д. Д., Діденко Ю. В. Програмування мовами С та С++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К.: , 2012. – 112 с.
17. Кормен Т., Лейзерсон Ч. Э., Ривест Р. Н., Штайн К. Алгоритмы: построение и анализ / – М. : Вильямс, 2015. – 1328 с.
18. Новотарський М.А. Алгоритми та методи обчислень: навч. посіб. для студ. спеціальностей 121 «Інженерія програмного забезпечення», спеціалізації «Програмне забезпечення високопродуктивних комп'ютерних систем та мереж» та 123 «Комп'ютерна інженерія», спеціалізації «Комп'ютерні системи та мережі» / М. А. Новотарський; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 4648 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 407 с.
19. Буч Г. UML. Руководство пользователя / Буч Г., Рамбо Дж., Джекобсон А. - М. : ДМК Пресс, 2003. - 432 с.: ил 18.
20. Дідковська М.В. Проектування програмного забезпечення засобами UML//Наукові вісті. - 2007, 16 с.
21. Положення про дипломні роботи(проекти) випускників Національного авіаційного університету СМЯ НАУ П 03.01(10) – 02 – 2017.

22. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

23. Дипломне проектування в комп'ютерній інженерії: навч.- метод. посібник / В. Д. Далека, А. І. Поворознюк., А. О. Подорожняк, Ю. І. Панфілов – Харків: НТУ «ХП», 2019. – 56 с.

ДОДАТОК А
Програмний код

```
#include<iostream>  
#include<random>  
#include<vector>  
#include<map>  
#include<fstream>  
#include<iterator>  
#include<time.h>  
  
using namespace std;  
  
//клас "Канал обслуговування"  
class ServiceChannel {  
    public:  
        int current_state; //поточний стан каналу обслуговування,  
        можливі стани: -1 - канал виведений з ладу, 0 - вільний, 1 - обробляє запит  
        bool state_changed; //логічне значення зміни стану каналу  
        bool disabled; //логічне значення виведеного з ладу каналу  
        int processed_requests; //кількість оброблених запитів каналом  
        int number_of_requests; //кількість запитів, які надійшли до  
        каналу, незалежно від того чи були вони прийняті, чи ні  
        int processing_time; //час, витрачений на оброблення запитів  
        каналом обслуговування  
        int waiting_time; //час очікування запиту каналом
```

```

int disabled_time; //час перебування каналу в стані виведеному з
ладу

//функція подачі запиту на канал обслуговування
bool submitting_request_to_channel(
    int &correct_request, //на канал надійшов коректний запит
    int &malicious_request, //на канал надійшов шкідливий
запит
    poisson_distribution<int> &correct_distribution, //метод
імітації розподілу Пуассона для інтенсивності обробки коректного запиту
    poisson_distribution<int> &disabled_distribution, //метод
імітації розподілу Пуассона для інтенсивності ремонту каналу, виведеного з
ладу
    default_random_engine &generator //генератор випадкових
чисел
){
    state_changed = false;

    if (correct_request) {
        //якщо надійшов коректний запит
        number_of_requests++; //збільшуємо кількість
запитів, які надійшли на канал
    }

    if (!disabled) {
        //якщо канал не виведений з ладу

        if (current_state != 0) {
            //якщо канал зайнятий

```

```

if (current_state == 1) {
    //якщо канал обробляє коректний запит
    processing_time++; //нідраховуємо час
    обробки запиту

    int result_of_processing_request =
correct_distribution(generator); //обробка коректного запиту, результат
залежить від інтенсивності обслуговування каналу в залежності від якої
будується розподіл Пуассона

    if (result_of_processing_request) {
        //якщо обробка завершилася
        current_state = 0; //стан каналу
        вільний

        state_changed = true; //стан
        змінився
    }
}
else {
    //якщо канал обробляє шкідливий запит
    disabled_time++; //нідраховуємо час
    перебування каналу у стані виведеному з ладу

    int result_of_channel_repair =
disabled_distribution(generator); //ремонт каналу, результат залежить від
інтенсивності ремонту каналу в залежності від якої будується розподіл
Пуассона

    if (result_of_channel_repair) {
        //якщо ремонт завершився
        current_state = 0; //стан каналу
        вільний
    }
}

```

```

state_changed = true; //стан
змінився
    }
}
}
else if (correct_request > 0 || malicious_request > 0) {
//якщо канал вільний і надійшов коректний або
шкідливий запит
state_changed = true; //стан змінився

if (!(correct_request > 0 && malicious_request >
0)) {
//якщо запит або коректний, або
шкідливий

if (correct_request > 0) {
//якщо це коректних запит
current_state = 1; //канал
знаходиться в стані обробки запиту
processed_requests++; //збільшуємо
кількість оброблених запитів
correct_request--; //зменшуємо
кількість коректних запитів, які надійшли, оскільки один із них щойно
обробили
}
else {
//якщо це шкідливий запит
current_state = -1; //канал
знаходиться в стані виведеного з ладу

```

```

        malicious_request--; //зменшуємо
кількість шкідливих запитів, які надійшли, оскільки один із них щойно
обробили

        disabled = true; //канал виведено з
ладу

    }

    return true;
}

else {
    //якщо одночасно надійшли і коректний, і
шкідливий запити

    int select_request = rand() % 2;
//випадковим чином обираємо який з запитів почне оброблятися каналом

    if (select_request == 0) {
        //якщо обрано шкідливий запит
        current_state = -1; //канал
знаходиться в стані виведеного з ладу

        malicious_request--; //зменшуємо
кількість шкідливих запитів, які надійшли, оскільки один із них щойно
обробили

        disabled = true; //канал виведено з
ладу

    }

    else {
        //якщо обрано коректний запит
        current_state = 1; //канал
знаходиться в стані обробки запиту

```

```

        processed_requests++; //збільшуємо
кількість оброблених запитів
        correct_request--; //зменшуємо
кількість коректних запитів, які надійшли, оскільки один із них щойно
обробили
    }

    return true;
}
}
else {
    //якщо не надійшло ні одного запиту
    waiting_time++; //збільшуємо час очікування
запиту каналом
    return false;
}
}
else {
    //якщо канал виведений з ладу
    disabled_time++; //збільшуємо час перебування
каналу у стані виведеному з ладу
    return false;
}
}
}

```

```

ServiceChannel() {
    current_state = 0;
    processed_requests = 0;
    number_of_requests = 0;
    processing_time = 0;
}

```



```

        waiting_time = 0;
        disabled_time = 0;
    }
    ~ServiceChannel() {};
};

//клас "Система обслуговування"
class ServiceSystem {
    public:
        vector<ServiceChannel> channels; //вектор із множини класів
ServiceChannel
        vector<vector<int>> state_history; //вектор із векторів, які
зберігають у собі стани системи в кожен момент часу, відображають
динаміку зміни станів системи
        map<vector<int>, int> state_statistics; //вектор із всіх можливих
станів, в яких побувала система і кількість часу проведеного в кожному з
цих режимів
        int time; //час роботи системи
        int number_of_lost_correct_requests; //кількість втрачених
коректних запитів
        int number_of_processed_correct_requests; //кількість оброблених
коректних запитів

        //визначаємо кількість каналів обслуговування
        ServiceSystem(int number_of_channels) {
            channels.resize(number_of_channels, ServiceChannel());
            number_of_lost_correct_requests = 0;
            number_of_processed_correct_requests = 0;
        }
};

```

```

//функція, яка описує роботу системи та вхідний потік запитів
void run(
    double flow_of_correct_requests,
    double flow_of_malicious_requests,
    double processing_flow,
    double repair_flow,
    int _time
) {
    time = _time;
    vector<int> current_state(channels.size(), 0); //встановлюємо
    для всіх каналів початковий стан 0 (канал вільний)
    state_history.push_back(current_state);
    int length = 0;

    ofstream out("process.txt");
    ofstream out_s("sequence.txt");

    default_random_engine generator;
    //генеруються об'єкти, які підпорядковуються розподілу
    Пуассона з заданими інтенсивностями
    poisson_distribution<int>
    distribution_correct(flow_of_correct_requests);
    poisson_distribution<int>
    distribution_malicious(flow_of_malicious_requests);
    poisson_distribution<int>
    distribution_processing(processing_flow);
    poisson_distribution<int> distribution_repair(repair_flow);

    int count_correct = 0, count_malicious = 0;
    for (int i = 0; i < time; i++) {

```

```

        //отримуємо стан потоку запитів в даний момент
        часу в залежності від інтенсивності надходження запитів
        int correct = distribution_correct(generator);
        int malicious = distribution_malicious(generator);
        count_correct += correct;
        count_malicious += malicious;

        out_s << "correct: " << correct << ", malicious: " <<
malicious << endl; //виводимо стан потоку до файлу
        length++;

        bool state_changed = false;
        bool processed = false;

        //подаємо цей стан потоку на канал обслуговування и
        проводимо з ним необхідні операції
        for (int c = 0; c < channels.size(); c++) {
            bool processed_request =
channels[c].submitting_request_to_channel(
                correct,
                malicious,
                distribution_processing,
                distribution_repair,
                generator
            ); //подача запиту на канал
        }
        обслуговування

        if (channels[c].state_changed) {
            //якщо стан був змінений
            state_changed = true;
        }
    
```

```

                                current_state[c]
channels[c].current_state;
                                }
                                }

                                if (state_changed) {
                                    //якщо стан змінився
                                    for (int j = 0; j < state_history.back().size(); j++) {
                                        out.width(3);
                                        out << state_history.back()[j]; //виводимо
стани системи
                                    }

                                        out.width(7);
                                        out << length << " it = "; //виводимо час
перебування системи у цих станах
                                        out.width(11);
                                        out << double(length) / time * 100 << "%" <<
endl; //виводимо відсоткове співвідношення часу перебування системи у цих
станах до загального часу системи
                                        state_statistics[state_history.back()] += length;
//записуємо час перебування в даному стані
                                        state_history.push_back(current_state);
//записуємо даний стан системи
                                        length = 0; //обнуляємо час для наступних
підрахунків
                                    }

                                if (correct > 0) {
                                    //якщо надійшов коректний запит

```

```

        for (int c = 0; c < channels.size(); c++) {
            channels[c].disabled = false; //кожен з
каналів не вийде з ладу
        }
    }
}

state_statistics[state_history.back()] += length;
state_history.push_back(current_state);
length = 0;

//записуємо в файл statistics.txt статистику роботи
системи

out = ofstream("statistics.txt");
out << "##### STATISTICS
#####" << endl << endl;
map<vector<int>, int>::iterator cur;

//виводимо в файл стани системи та час, який вона
перебувала у цих станах
for (cur = state_statistics.begin(); cur != state_statistics.end();
cur++) {

    for (int i = 0; i < (*cur).first.size(); i++) {
        out.width(3);
        out << (*cur).first[i];
    }

    out.width(10);
    out << (*cur).second << " it = ";
    out.width(7);

```

```

        out << double((*cur).second) / time * 100 << "%" <<
endl;
    }

    //продуктивність кожного каналу обслуговування
    out << endl << "correct requests for channel:" << endl;
    double summ = 0;
    for (int i = 0; i < channels.size(); i++) {
        out << "channel " << i + 1 << ": processed requests = "
<< channels[i].processed_requests << ", "
            << double(channels[i].processed_requests) / time
<< " for 1 it, "
            << "number of correct requests = " <<
channels[i].number_of_requests << endl;
        summ += double(channels[i].processed_requests) / time;
        number_of_processed_correct_requests +=
channels[i].processed_requests;
    }

    number_of_lost_correct_requests = count_correct -
number_of_processed_correct_requests;

    out << endl << "TOTAL SYSTEM PERFORMANCE = " <<
summ << endl; //загальна продуктивність системи
    //кількість оброблених і втрачених запитів
    out << endl << "TOTAL CORRECT REQUESTS WERE
PROCESSED = " << number_of_processed_correct_requests << " = "
            << double(number_of_processed_correct_requests) /
(number_of_lost_correct_requests + number_of_processed_correct_requests) *
100 << "%" << "\t" << endl;

```

```

        out << endl << "TOTAL CORRECT REQUESTS WERE LOST
= " << number_of_lost_correct_requests << " = "
        <<      double(number_of_lost_correct_requests)      /
(number_of_lost_correct_requests + number_of_processed_correct_requests) *
100 << "%" << "\t" << endl;
    }

    ~ServiceSystem() {}
};

int main() {
    srand(time(NULL));
    //задання показників системи і кількість каналів обслуговування
    double lambda = 0.138, a = 0.9;
    double correct_flow = a * lambda, malicious_flow = (1 - a)*lambda;
    double processing_flow = 0.025, repair_flow = 0.05;
    int time = 200000;
    ServiceSystem system(2);
    system.run(correct_flow, malicious_flow, processing_flow, repair_flow,
time);
}

```