

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»
КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Курсова робота з освітнього компоненту
«Технології паралельних обчислень. Курсова робота»

**Тема: Імітаційна модель системи процесорів на основі
формального опису мережею Петрі**

Керівник:

ст.викл. Дифучина Олександра Юріївна

«Допущено до захисту»

«___» _____ 2024 р.

Захищено з оцінкою

Члени комісії:

Виконавець:

Панченко Сергій Віталійович

студент групи ІП-11

залікова книжка № ІП-1123

«5» грудня 2024 р.

Інна СТЕЦЕНКО

Олександра ДИФУЧИНА

Київ – 2024

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Кафедра інформатики та програмної інженерії

Дисципліна «Моделювання систем»

Спеціальність 121 Інженерія програмного забезпечення

Курс 4 Група ІП-11 Семестр 1

ЗАВДАННЯ

на курсову роботу студента

Панченка Сергія Віталійовича

(прізвище, ім'я, по батькові)

1. Тема роботи «Імітаційна модель системи процесорів на основі формального опису мережею Петрі»

2. Термін здачі студентом закінченої роботи "5" грудня 2024р.

3. Зміст розрахунково-пояснювальної записки

1. Опис 2. Псевдокод 3. Реалізація 4. Реалізація 5. Реалізація 6. Проведення експериментів над моделями. Висновки.

4. Дата видачі завдання "29" жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів виконання курсової роботи | Термін виконання етапів роботи | Примітка |
|---|-----------------------------------------------------------------------------|--------------------------------|----------|
| 1 | Отримання індивідуального завдання на курсову роботу | 29.10.2024 | |
| 2 | Розробка концептуальної моделі системи | 29.11.2024 | |
| 3 | Розробка формалізованої моделі системи | 03.11.2024 | |
| 4 | Алгоритмізація моделі системи та її програмна реалізація | 08.11.2024 | |
| 5 | Експериментальне дослідження моделі системи | 13.11.2024 | |
| 6 | Інтерпритація результатів моделювання, формулювання висновків та пропозицій | 15.11.2024 | |
| 7 | Оформлення пояснювальної записки | 17.11.2024 | |
| 8 | Захист КР | 5.12.2024 | |

Студент _____ Панченко С. В.
(підпис)

Керівник _____ Дифучина О.Ю.
(підпис)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка курсової роботи складається з 5 розділів, містить 2 рисунки, 2 таблиці, 1 додаток, 2 джерела.

Мета: розробити імітаційну модель системи процесорів на основі формального опису мережею Петрі.

У розділі розробки концептуальної моделі було окреслено концептуальну модель та визначено необхідні властивості імітаційних алгоритмів.

У розділі аналіз розробки формалізованої моделі було описано моделі у рамках формалізму мереж Петрі.

У розділі алгоритмізації моделі та її реалізації було описано усі деталі щодо алгоритму та його реалізації.

У розділі експериментів на моделі було проведено верифікацію моделі та факторний експеримент.

У розділі інтерпретації результатів моделювання та експериментів було описано отримані результати в рамках проведеного моделювання та експериментування.

КЛЮЧОВІ СЛОВА: МОДЕЛЮВАННЯ СИСТЕМ, МЕРЕЖІ ПЕТРІ, БАГАТОПРОЦЕСОРНА ОБЧИСЛЮВАЛЬНА СИСТЕМА.

ЗМІСТ

| | |
|---------------------------------------------------------------|----|
| Постановка завдання..... | 6 |
| Вступ..... | 8 |
| 1 Розробка концептуальної моделі..... | 10 |
| 1.1 Опис задачі..... | 10 |
| 1.2 Структурна схема моделі..... | 10 |
| 1.3 Опис процесу..... | 10 |
| 1.4 Вхідні змінні..... | 11 |
| 1.5 Вихідні змінні..... | 11 |
| 1.6 Функціональна залежність..... | 12 |
| 1.7 Цільова функція..... | 12 |
| 1.8 Обмеження..... | 12 |
| 1.9 BPNM-діаграма..... | 12 |
| 1.10 Висновки до розділу..... | 14 |
| 2 Розробка формалізованої моделі..... | 15 |
| 2.1 Вхідні параметри..... | 16 |
| 2.2 Вихідні параметри..... | 18 |
| 2.3 Діаграма..... | 20 |
| 2.4 Висновок до розділу..... | 22 |
| 3 Алгоритмізація моделі та її реалізація..... | 23 |
| 4 Експерименти на моделі..... | 24 |
| 5 Інтерпретація результатів моделювання та експериментів..... | 25 |
| Висновки..... | 26 |
| Список використаних джерел..... | 27 |
| Додатки..... | 28 |

ПОСТАНОВКА ЗАВДАННЯ

Багатопроцесорна обчислювальна система складається з двох процесорів із загальною оперативною пам'яттю обсягом 131 сторінка, чотирьох накопичувачів на дисках, кожний із яких доступний обом процесорам, і одного каналу передачі даних. Завдання надходять у систему із середньою інтенсивністю, рівною 12 завданням у хвилину відповідно до розподілу Пуассона. Загальний час, необхідний процесору на обробку завдання, розподілено нормально з математичним сподіванням 10 секунд та середнім квадратичним відхиленням 3 секунди. Час обробки процесором включає переривання, необхідні для здійснення обміну по каналу вводу-виводу. Інтервали між перериваннями розподілені за негативний експоненціальний розподілом з математичним сподіванням, що дорівнює оберненій величині середньої інтенсивності операцій вводу-виводу завдання. Середня інтенсивність операцій введення-виведення розподілена рівномірно на інтервалі від 2 секунд до 10 секунд. Операції введення-виведення призначаються конкретному диску. Завданню, що надходить у систему, призначається пріоритет, що є величиною, оберненою до потреби в пам'яті. Потреба завдання в пам'яті розподілена рівномірно в інтервалі від 20 до 60 сторінок. Як тільки пам'ять виділена для завдання, один з вільних процесорів починає його обробку. При видачі запиту на здійснення введення-виведення завдання може продовжувати використання процесора доти, доки в черзі залишиться тільки один запит. Таким чином, якщо зроблений запит на здійснення введення-виведення і один запит вже очікує в черзі, то процесор звільняється, а запит на введення-виведення розміщується в черзі. Після виконання поточного запиту введення-виведення процесор може відновити обробку завдання в тому випадку, якщо вона вільна. Після переривання процесора автоматично виконується запит введення-виведення з призначеним завданню диском. Таким чином, здійснюється прямий доступ до диска з процесора. Передбачається, що час позиціонування диска розподілено рівномірно на інтервалі від 0,0 до 0,075 секунд. Одночасно може здійснюватися

тільки одна операція позиціонування диска. Після позиціонування здійснюється обмін даними по каналу передачі даних. Час обміну дорівнює $0,001 \times (2,5 + h)$, де h - рівномірно розподілена на інтервалі від 0 до 25 величина. Після здійснення обміну запит введення-виведення вважається виконаним. Визначити загальний час виконання завдання в системі, а також статистичні оцінки завантаження усіх чотирьох дисків, каналу передачі даних та обох процесорів. Крім того, необхідно одержати оцінку середнього використання пам'яті, статистику щодо кількості завдань, які очікують виділення ресурсу, та щодо часу очікування.

ВСТУП

У сучасному світі багатопроцесорні обчислювальні системи відіграють ключову роль у вирішенні складних обчислювальних задач. Ефективність їх роботи залежить від багатьох факторів, включаючи управління пам'яттю, розподіл процесорного часу та організацію введення-виведення. Тому дослідження характеристик таких систем є актуальним завданням для оптимізації їх роботи та підвищення продуктивності.

Метою даної курсової роботи є дослідження характеристик багатопроцесорної обчислювальної системи, що складається з двох процесорів із спільною оперативною пам'яттю, чотирьох дискових накопичувачів та одного каналу передачі даних. Особлива увага приділяється аналізу завантаженості системних ресурсів та часу виконання завдань.

Для досягнення поставленої мети використовується апарат мереж Петрі, який є потужним математичним інструментом для моделювання та аналізу паралельних та розподілених систем. Мережі Петрі дозволяють ефективно описувати та досліджувати асинхронні, паралельні процеси в системі, враховуючи стохастичну природу процесів, що відбуваються в ній. У роботі застосовуються різні типи ймовірнісних розподілів для моделювання надходження завдань (розподіл Пуассона), часу обробки завдань (нормальний розподіл), інтервалів між перериваннями (експоненційний розподіл) та інших характеристик системи.

Для реалізації моделі використовується спеціалізоване програмне забезпечення PetriObjModelPaint, розроблене для моделювання та аналізу мереж Петрі. Цей інструмент надає можливості для створення, візуалізації та дослідження мережевих моделей, а також дозволяє отримувати статистичні оцінки завантаження всіх компонентів системи, включаючи процесори, диски, канал передачі даних, аналізувати використання пам'яті та характеристики черг завдань.

Результати дослідження дозволять оцінити ефективність роботи системи та виявити потенційні "вузькі місця" в її функціонуванні, що може бути використано для подальшої оптимізації роботи багатопроцесорних обчислювальних систем.

1 РОЗРОБКА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ

У цьому розділі представлено концептуальну модель багатопроцесорної обчислювальної системи, яка включає опис її складових, основних процесів, правил взаємодії елементів та параметрів, необхідних для подальшого моделювання і аналізу.

1.1 Опис задачі

Ціль моделювання – визначити показники роботи багатопроцесорної обчислювальної системи, зокрема:

- загальний час виконання завдань;
- завантаження дисків, каналу передачі даних і процесорів;
- середнє використання пам'яті;
- кількість завдань, що очікують на виділення ресурсу;
- час очікування виділення пам'яті.

1.2 Структурна схема моделі

Об'єктом моделювання є багатопроцесорна система, яка включає:

- два процесори, що використовують спільну оперативну пам'ять на 131 сторінку;
- чотири диски, доступні обом процесорам;
- один канал передачі даних, який використовується для обміну між процесорами та дисками.

1.3 Опис процесу

Процес складається з таких елементів:

- завдання надходять у систему з інтенсивністю 12 завдань/хв за розподілом Пуассона;
- кожному завданню призначається пріоритет, обернений до його потреби в пам'яті;

- потреба завдань у пам'яті розподілена рівномірно в межах 20–60 сторінок;
- якщо пам'ять доступна, одразу призначається один із вільних процесорів;
- час виконання завдання розподілений нормально із середнім 10 секунд та відхиленням 3 секунди;
- у процесі виконання завдань виникають переривання для обміну даними; інтервали між перериваннями розподілені експоненційно з параметром, оберненим до інтенсивності операцій введення-виведення (2–10 секунд);
- операції введення-виведення включають;
- позиціонування диска ($0-0,075$ с);
- передачу даних ($0,001 \times (2,5+h)$), де h розподілено рівномірно в межах 0–25);
- завдання звільняє процесор, якщо черга операцій введення-виведення складається з максимум одного завдання; в іншому випадку, воно продовжує займати процесор.

1.4 Вхідні змінні

Модель має такі вхідні змінні, як-от:

- інтенсивність надходження завдань (12 завдань/хв);
- потреба завдань у пам'яті (20–60 сторінок);
- час виконання завдання процесором (нормальний розподіл: середнє 10 с, відхилення 3 с);
- інтенсивність операцій введення-виведення (2–10 с);
- час позиціонування диска ($0-0,075$ с);
- час передачі даних ($0,001 \times (2,5+h)$), h у межах 0–25).

1.5 Вихідні змінні

Модель має такі вихідні змінні, як-от:

- загальний час виконання завдань;

- завантаження процесорів, дисків, каналу передачі даних;
- середнє використання пам'яті;
- кількість завдань, що очікують виділення пам'яті чи ресурсів;
- час очікування виділення пам'яті.

1.6 Функціональна залежність

Змінюючи параметри системи, наприклад, обсяг оперативної пам'яті, можна впливати на завантаження ресурсів і середній час виконання завдань.

1.7 Цільова функція

Оптимізація завантаження системи, зменшення часу очікування ресурсів і максимізація продуктивності.

1.8 Обмеження

Модель має такі обмеження, як-от:

- оперативна пам'ять виділяється завданню на весь час виконання;
- одночасно може виконуватись лише одна операція позиціонування диска;
- канал передачі даних є спільним ресурсом.

1.9 BPMN-діаграма

BPMN-діаграма (Business Process Model and Notation[1]) – це графічне представлення процесів, яке дозволяє детально описати їхню послідовність, взаємодію елементів системи та потік даних між ними. Вона використовується для візуалізації концептуальної моделі, допомагаючи зрозуміти логіку роботи системи, виявити можливі вузькі місця та полегшити подальшу формалізацію і моделювання.

На рисунку 1.1 представлено BPMN-діаграму моделі, яка відображає ключові етапи обробки завдань і взаємодію різних компонентів системи через обмін повідомленнями. Діаграма ілюструє потік даних і послідовність дій від генерації завдання до його завершення.

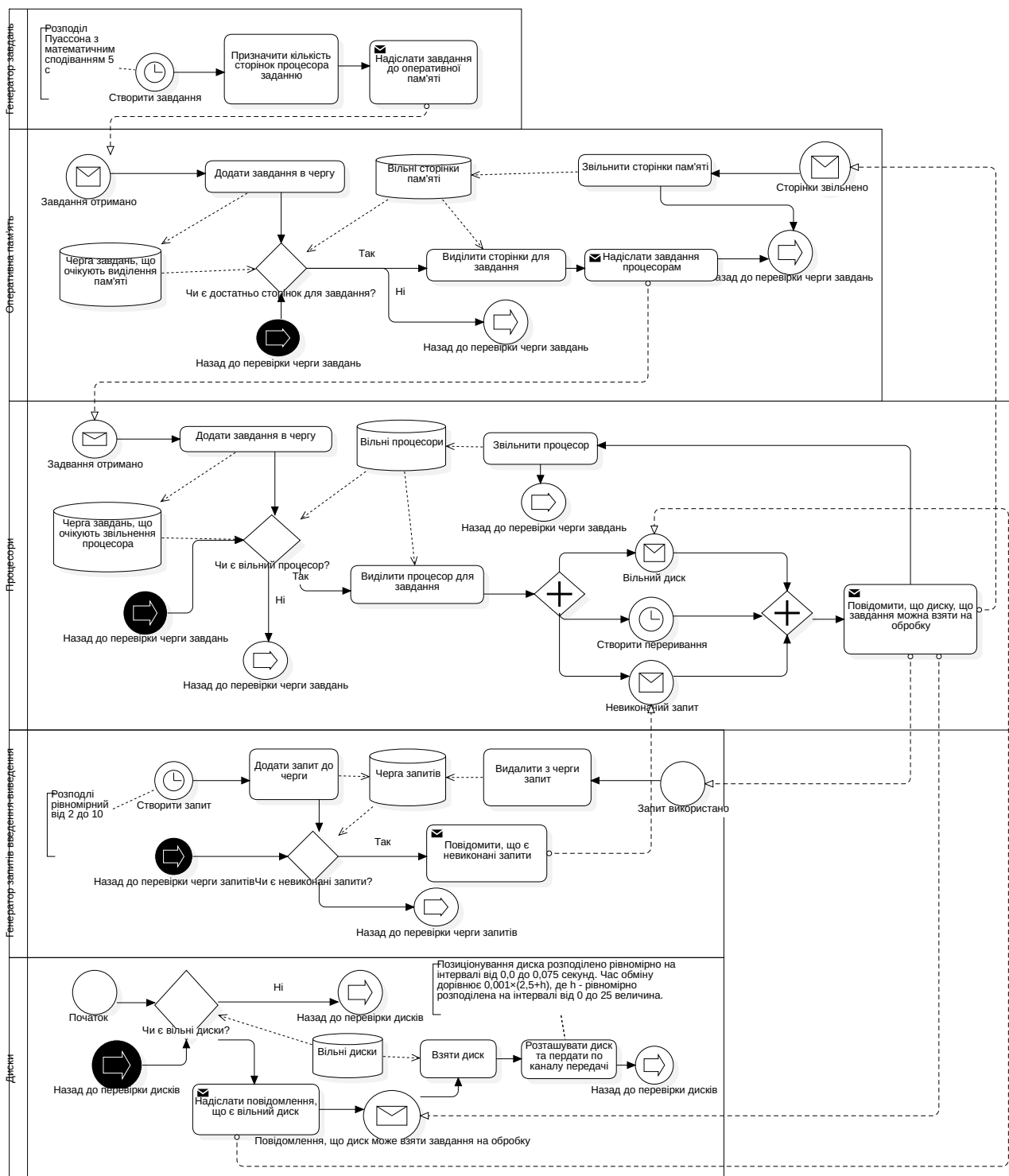


Рисунок 1.1 — BPMN-діаграма

Розглянемо блок генерації завдань. Через визначений проміжок часу, вказаний у коментарі, відбувається подія створення нового завдання. Після

цього завданню призначається обсяг пам'яті (кількість сторінок), необхідний для його обробки, і воно передається в оперативну пам'ять.

Оперативна пам'ять отримує повідомлення про нове завдання, додає його в чергу й перевіряє наявність вільних сторінок для виконання хоча б одного завдання з черги. Щойно вільна пам'ять стає доступною, вона виділяється, і завдання надсилається до процесорів.

Процесори приймають повідомлення про нове завдання і додають його в чергу. Як тільки один із процесорів стає вільним, він починає обробку завдання. По завершенню обробки перевіряються три умови: наявність вільного диска, невиконаний запит введення-виведення та отримання відповідних повідомлень. Коли ці умови виконано, процесор звільняється, а система надсилає кілька повідомлень: генератору запитів про виконання запиту, оперативній пам'яті про звільнення сторінок і диску про готовність прийняти завдання.

Розглянемо блок генератора запитів. Через заданий час відбувається подія створення нового запиту, який додається в чергу. Далі постійно перевіряється наявність хоча б одного невиконаного запиту в черзі. Якщо такий запит є, повідомлення надсилається процесорам.

Коли диск отримує повідомлення про готовність захопити завдання, він розпочинає позиціонування своєї головки зчитування, після чого виконується передача даних через канал введення-виведення. Завдання вважається завершеним після успішного виконання цих операцій.

1.10 Висновки до розділу

У висновку можна зазначити, що розроблена концептуальна модель багатопроцесорної обчислювальної системи забезпечує базу для формалізації процесів, побудови алгоритму імітації та проведення експериментального дослідження. Визначені вхідні змінні, параметри та вихідні характеристики дозволяють ефективно аналізувати роботу системи та оптимізувати її продуктивність.

2 РОЗРОБКА ФОРМАЛІЗОВАНОЇ МОДЕЛІ


У цьому розділі представлено формалізовану модель багатопроцесорної обчислювальної системи, розроблену з використанням мережі Петрі. Модель описує елементи системи, такі як пам'ять, процесори, диски та канал передачі даних, а також події, що визначають їх взаємодію. Вказано числові параметри для кожного компонента, правила спрацьовування переходів, а також описано генерацію випадкових величин для вхідних змінних. У розділі також подано формули для обчислення вихідних характеристик системи, які дозволяють аналізувати її продуктивність, завантаженість та ефективність роботи. Створена модель є основою для подальшого алгоритму імітації та експериментального дослідження.


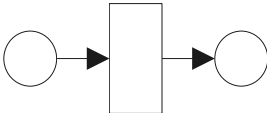



Мережа Петрі – це математичний апарат для моделювання дискретних систем, який складається з позицій, переходів та дуг. Позиції представляють стани системи, переходи – події або дії, що змінюють ці стани, а дуги визначають зв'язки між позиціями і переходами. Стан мережі визначається кількістю маркерів у позиціях, які називаються токенами [2][с. 67].

В основі роботи мережі лежить правило спрацьовування переходу: якщо у всіх вхідних позиціях переходу кількість маркерів не менша за кратність відповідних дуг, перехід активується. Під час спрацьовування маркери переміщуються з вхідних позицій до вихідних відповідно до кратності дуг. Мережа Петрі дозволяє описувати процеси з конфліктними переходами, багатоканальні операції та стохастичні події, що забезпечує її високу гнучкість для моделювання складних систем.

У таблиці 2.1 наведено елементи формалізації мережі Петрі [2][с. 69].

Таблиця 2.1 Мережа Петрі

| Назва | Позначення | Опис |
|---------|-------------------------------------------------------------------------------------|----------------|
| Перехід |  | Позначає подію |

| Назва | Позначення | Опис |
|--------------|-------------------------------------------------------------------------------------|---------------------------------------------|
| Позиція |  | Позначає умову |
| Дуга |  | Позначає зв'язки між подіями та умовами |
| Маркер(один) |  | Позначає виконання (або не виконання) умови |
| Багато фішок |  | Позначає багатократне виконання умови |
| Багато дуг |  | Позначає велику кількість зв'язків |

2.1 Вхідні параметри

Розглянемо параметри переходів моделі у таблиці 2.2. Нехай маємо відрізок $[M; N]$, що позначає діапазон можливої кількості сторінок, що завдання може займати. Нехай $K \in [M; N]$, де K — кількість сторінок, що займає довільне завдання.

Таблиця 2.2 Параметри переходів

| Назва переходу | Часова затримка | Значення пріоритету | Значення ймовірності запуску |
|-----------------------|-----------------|---------------------|------------------------------|
| generate_task | Poisson(5.0) | 0 | 1 |
| generate_task_K_pages | 0 | 0 | $\frac{1}{M - N}$ |

| Назва переходу | Часова затримка | Значення пріоритету | Значення ймовірності запуску |
|----------------------------------|----------------------|---------------------|------------------------------|
| fail_allocate_task_K_pages | 0 | -1 | 1 |
| try_allocate_task_K_pages | 0 | 0 | 1 |
| wait_allocate_task_20_pages | 0 | 0 | 1 |
| process_task_K_pages | Norm(10, 3) | $N - K$ | 1 |
| generate_io_request | Unif(2, 10) | 0 | 1 |
| create_io_task_20_pages | 0 | $N - K$ | 1 |
| take_up_disk_task_K_pages | 0 | $N - K$ | 1 |
| generate_interrupt_K_pages | Exp(6) | 0 | 1 |
| drop_interruprt | 0 | -1 | 1 |
| place_disk_K_pages | Unif(0, 0.075) | $N - K$ | 1 |
| io_channel_transfer_task_K_pages | Unif(0.0025, 0.0275) | $N - K$ | 1 |

Розглянемо параметри позицій у таблиці 2.3.

Таблиця 2.3 Параметри позицій

| Назва позиції | Початкове значення |
|----------------|--------------------|
| generator_task | 1 |

| Назва позиції | Початкове значення |
|----------------------------------|--------------------|
| generated_task | 0 |
| task_K_pages | 0 |
| fail_allocate_token_task_K_pages | 0 |
| allocated_task_K_pages | 0 |
| total_wait_allocate_task | 0 |
| pages | 131 |
| generator_io_request | 1 |
| processors | 2 |
| generated_request | 0 |
| io_task_K_pages | 0 |
| generator_interrupt | 1 |
| generated_interrupt | 0 |
| busy_disk_task_K_pages | 0 |
| free_disks | 4 |
| disk_placed_task_K_pages | 0 |
| is_disk_placement_available | 1 |
| finished_tasks_task_K_pages | 0 |
| finished_tasks | 0 |

2.2 Вихідні параметри

Для початку визначимо загальний час виконання завдання в системі. Для цього зафіксуємо моменти часу, коли відбувається вихід з переходів generate_task_K_pages та finish_task_K_pages. Тоді ми, маючи різницю цих двох моментів часу, можемо визначити час, що провело певне завдання у системі.

Далі можна знайти суму усіх відрізків часу, розділити її на кількість виконаних завдань та отримати середній витрачений завданням час в системі.

$$\Delta T_{\text{average time in system}} = \frac{\sum_{K=M}^N \left[\sum_{i=0}^{I_{K \text{ pages}}} [T_{K \text{ pages, finish, } i} - T_{K \text{ pages, start, } i}] \right]}{\sum_{K=M}^N I_{K \text{ pages}}},$$

де N - максимальна кількість сторінок, M - мінімальна кількість сторінок,
I - кількість завершених з кількістю сторінок K.

Час очікування на виділення пам'яті розраховується аналогічно:

$$\Delta T_{\text{average wait allocation}} = \frac{\sum_{K=M}^N \left[\sum_{i=0}^{I_{K \text{ pages}}} [T_{K \text{ pages, wait alloc, } i} - T_{K \text{ pages, fail alloc, } i}] \right]}{\sum_{K=M}^N I_{K \text{ pages}}}$$

Завантаження каналу передачі є результат ділення часу, що він працював, до всього часу моделювання. Аналогічно визначається завантаження усіх чотирьох дивсків та процесорів.

$$L_{\text{processors}} = \frac{\sum_{K=M}^N \left[\sum_{i=0}^{I_{\text{process task, } K, \Delta T}} \Delta T_{\text{process task, } K, i} \right]}{\sum_{K=M}^N I_{\text{process task, } K, \Delta T}},$$

$$L_{\text{disks}} = \frac{\sum_{K=M}^N \left[\sum_{i=0}^{I_{\text{place disk, } K, \Delta T}} \Delta T_{\text{place disk, } K, i} \right]}{\sum_{K=M}^N I_{\text{place disk, } K, \Delta T}},$$

$$L_{\text{io channel}} = \frac{\sum_{K=M}^N \left[\sum_{i=0}^{I_{\text{io channel transfer, } K, \Delta T}} \Delta T_{\text{io channel transfer, } K, i} \right]}{\sum_{K=M}^N I_{\text{io channel transfer, } K, \Delta T}},$$

де $I_{\text{io channel transfer}}$, $I_{\text{place disk}}$, $I_{\text{process task}}$ - це кількість виходів з відповідних переходів.

2.3 Діаграма

На рисунку 2.1 можна розглянути діаграму мережі Петрі.

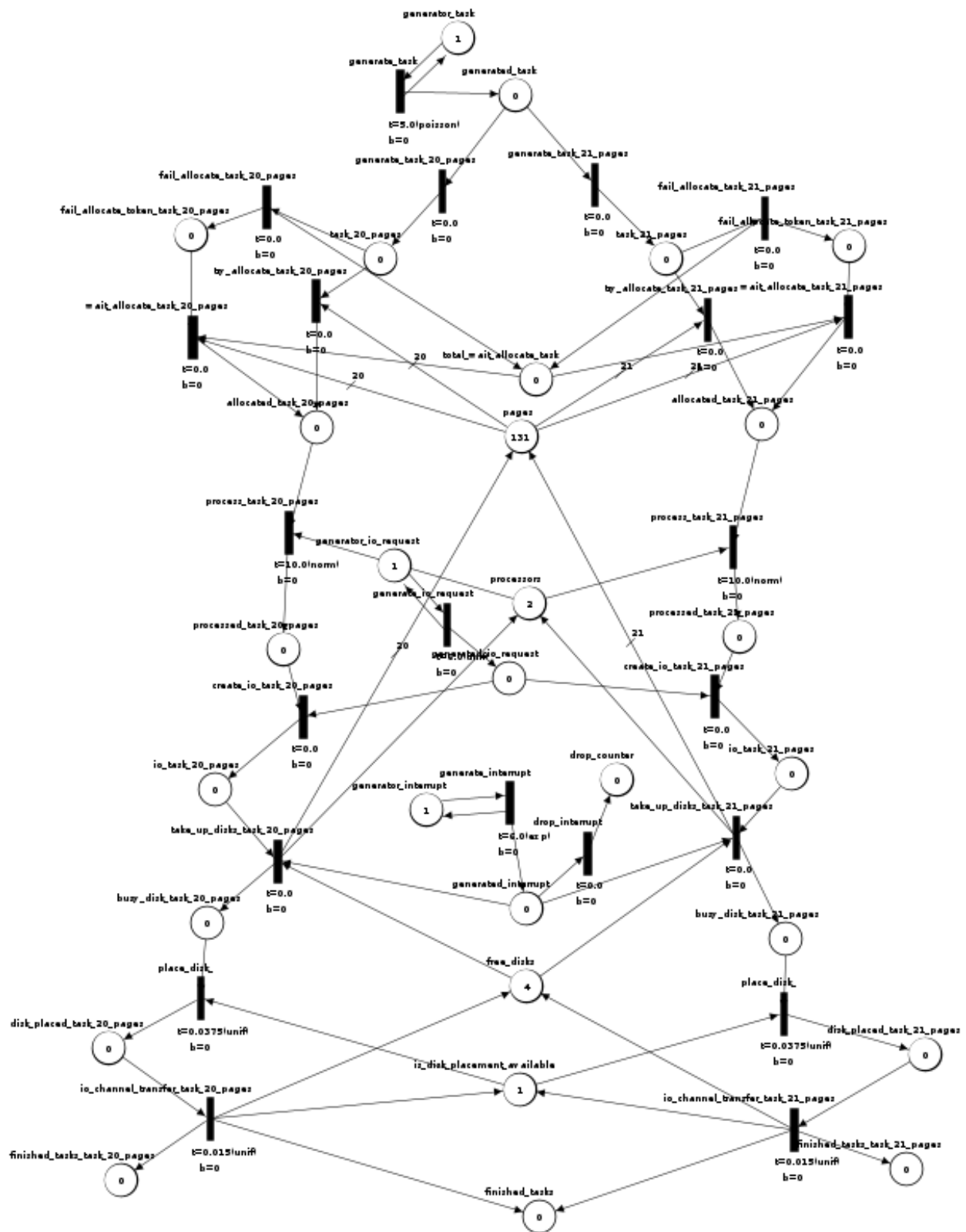


Рисунок 2.1 — Мережа Петрі з двома типами завдань

Для зручності вона побудована лише з двома типами завдань. Це робить її менш нагромадженою, оскільки частини схеми для кожного завдання є однаковими.

Основними компонентами мережі є генератор завдань, оперативна пам'ять, процесори, генератор запитів введення-виведення, диски, канал передачі даних та блок завершення роботи. Генератор завдань представлений позицією `generated_task`, яка містить токен, що активує перехід `generate_task`. Цей перехід генерує нові завдання з пуассонівським розподілом часу. Згенеровані завдання додаються в позиції, які відповідають їхньому обсягу пам'яті, наприклад, `task_20_pages` для завдань із потребою у 20 сторінок.

Оперативна пам'ять представлена позицією `pages`, яка відображає загальну кількість доступних сторінок (131 сторінка). Переходи `try_allocate` і `wait_allocate` перевіряють наявність вільної пам'яті та виділяють її для завдань відповідного обсягу. У разі успішного виділення токени передаються до позиції `allocated`, а у разі невдачі – у `fail_allocate_token`.

Процесори моделюються позицією `processors`, яка визначає кількість доступних процесорів (2 процесори). Переходи `process` відповідають за обробку завдань із використанням нормального розподілу часу.

Генератор запитів введення-виведення моделюється позицією `generated_io_request`, яка активує перехід `create_io`. Цей перехід генерує запити введення-виведення для завдань, які були оброблені процесорами.

Диски представлені позицією `free_disks`, яка відображає кількість доступних дисків (4 диски). Перехід `place_disk` імітує позиціонування головки зчитування з використанням рівномірного розподілу часу. Після виконання цієї операції токени передаються через канал введення-виведення.

Канал введення-виведення моделюється позицією `is_disk_placement_available`, яка визначає готовність каналу до передачі даних. Перехід `io_channel_transfer` відповідає за передачу даних із використанням рівномірного розподілу часу. Завершення роботи моделюється позицією

finished_tasks, яка накопичує виконані завдання. Це дозволяє аналізувати продуктивність системи та обчислювати необхідні показники.

Логіка роботи мережі ґрунтується на потоках токенів між позиціями через переходи. Завдання генеруються з певним пріоритетом і обсягом пам'яті, який визначається рівномірним розподілом (від 20 до 60 сторінок). Якщо завдання не отримало пам'ять, воно повертається до черги. Виділені завдання передаються до процесорів, які здійснюють їх обробку із врахуванням стохастичних затримок. Після обробки завдання надсилають запити на диски. Якщо диск доступний, він виконує операцію зчитування та передачу даних через канал введення-виведення. Завдання вважається завершеним після успішної передачі даних і повернення ресурсів системі, включаючи пам'ять, процесори та диски.

2.4 Висновок до розділу

У цьому розділі було створено формалізовану модель багатопроцесорної обчислювальної системи, яка дозволяє математично описати її компоненти та взаємодії. Модель побудована з використанням мережі Петрі, що забезпечує можливість детального моделювання процесів генерації, обробки завдань, запитів введення-виведення та управління ресурсами. Було визначено всі необхідні вхідні параметри, їх числові характеристики, а також формули для обчислення вихідних характеристик системи. Створена модель є основою для розробки алгоритму імітації та проведення експериментального дослідження ефективності роботи системи.

3 АЛГОРИТМІЗАЦІЯ МОДЕЛІ ТА ЇЇ РЕАЛІЗАЦІЯ

4 ЕКСПЕРИМЕНТИ НА МОДЕЛИ

5 ІНТЕРПРЕТАЦІЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ ТА ЕКСПЕРИМЕНТІВ

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1., Business Process Model and Notation,
2. І. В. Стеценко, Моделювання систем, 2011

ДОДАТКИ