НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Комп’ютерний практикум № 8

з дисципліни

“Моделювання систем”

Виконала:

студентка групи ІС-71

Вознюк О.В.

Перевірила:

старший викладач

Новікова П.А.

Київ-2020

**Лістинг програми**

https://github.com/oleksandravozniuk/SystemsModelling/tree/master/SystemsModelling8

**Отримані результати**

**1**

1. Розробити імітаційну модель двосторонньої передачі повідомлень між двома комунікаційними вузлами з використанням базової мережі Петрі (**40 балів**):

Згідно протоколу передачу одного повідомлення від вузла А до вузла В описують такі події: запит від А на передачу в В, позитивна відповідь від вузла В вузлу А, передача повідомлення з А в В, відправка повідомлення вузлом А, отримання повідомлення вузлом В, відправка сигналу про успішне отримання повідомлення вузлом В, відправка сигналу про успішне отримання повідомлення вузлом В.

Двостороння передача означає, що обмін повідомленнями відбувається одночасно в обох напрямках. Через те, що у вузлі може зберігатись тільки одне повідомлення, може виникати тупикова ситуація, коли два вузли здійснили відправку і «зависли» в очікуванні підтвердження отримання повідомлення іншим вузлом. Для запобігання тупикової ситуації, використовується керуючий сигнал, що надає дозвіл на відправку повідомлення тільки одному з двох вузлів.

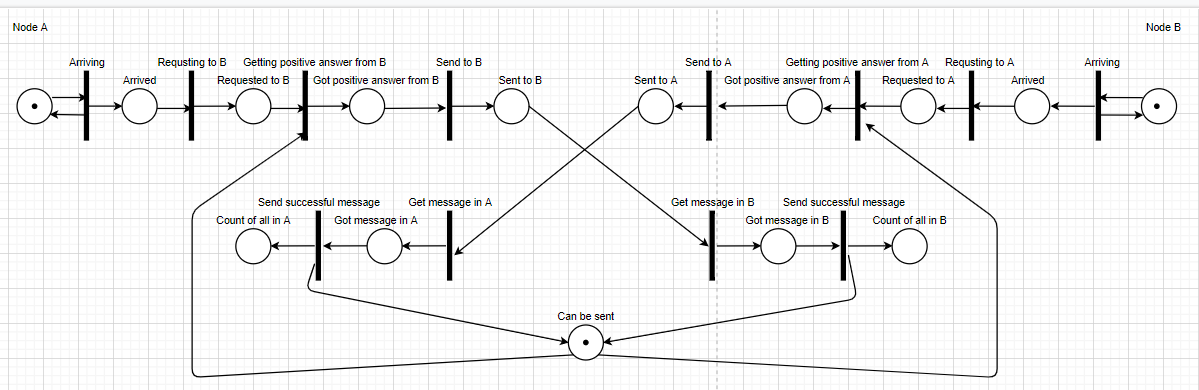


Рисунок 1- схема до завдання 1

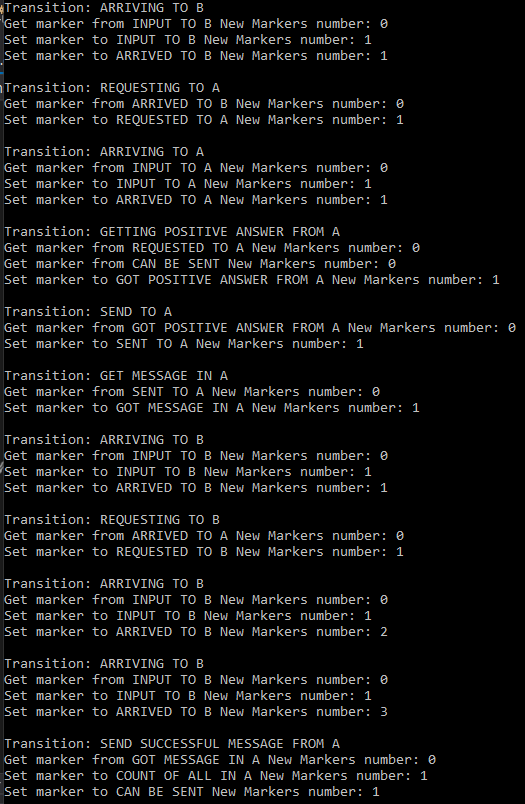


Рисунок 2 – протокол роботи алгоритму імітації для завдання 1

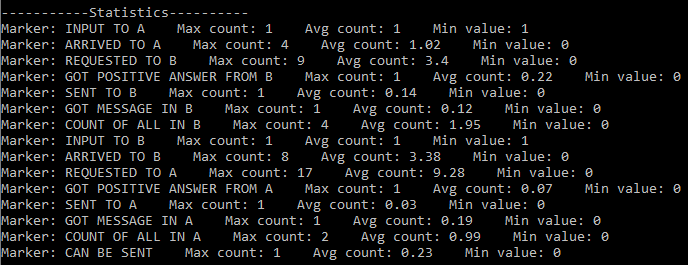


Рисунок 3 – результати роботи алгоритму імітації для завдання 1

**2**

2. Побудувати модель для задачі Producer – Consumer з використанням базової мережі Петрі (30 балів):

Процес Producer постачає задачі для виконання Consumer і розміщує їх в буфер (операція put). Процес Consumer виймає задачі з буфера (операція take) і обробляє їх. Оскільки буфер має обмеження n, то при досягненні максимального значення припиняється робота процесу Producer. Якщо задач в буфері немає, то робота процесу Consumer припиняється.

За результатами моделювання потрібно оцінити середнє значення буфера.

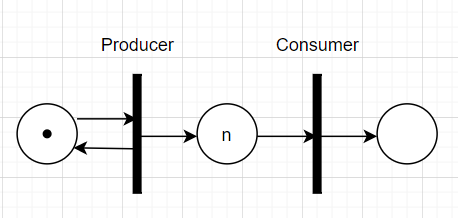


Рисунок 4 - схема до завдання 2

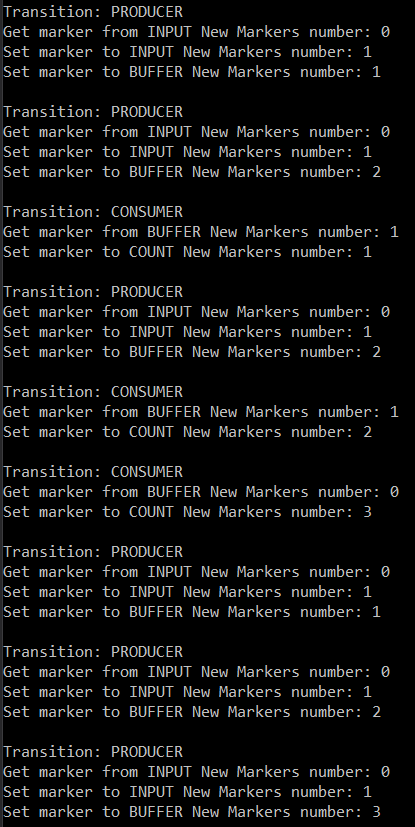


Рисунок 5 – протокол роботи алгоритму імітації для завдання 2

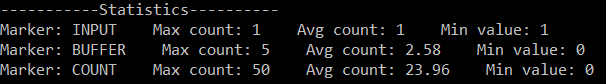


Рисунок 6 – результати роботи алгоритму імітації для завдання 2

**3**

3. Побудувати модель багатопроцесорної обробки завдань з використанням базової мережі Петрі (30 балів):

Задачі різних типів надходять у багатопроцесорну систему. Задачі першого типу вимагають усі процесори системи, задачі другого типу – третину всіх обчислювальних ресурсів, а задачі третього типу – половину ресурсів.

За результатами моделювання потрібно оцінити співвідношення кількості виконаних завдань.

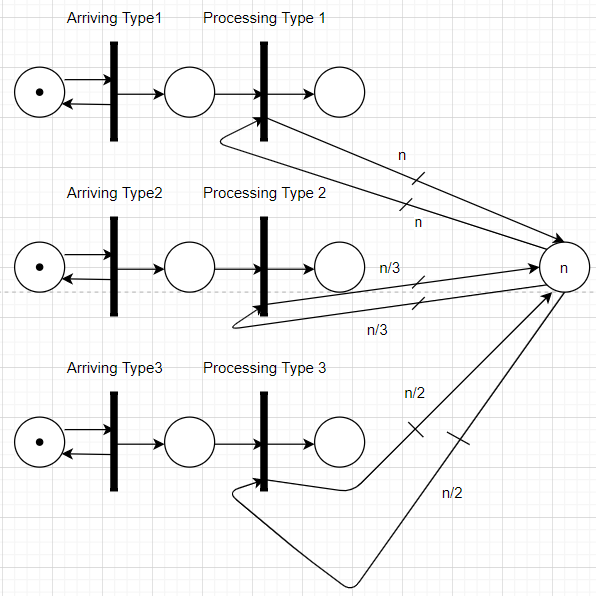


Рисунок 7- схема до завдання 3

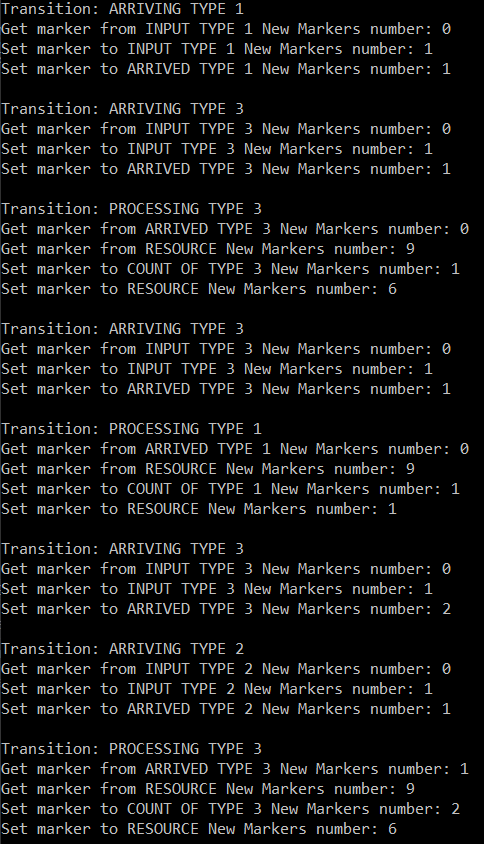


Рисунок 8 – протокол роботи алгоритму імітації для завдання 3

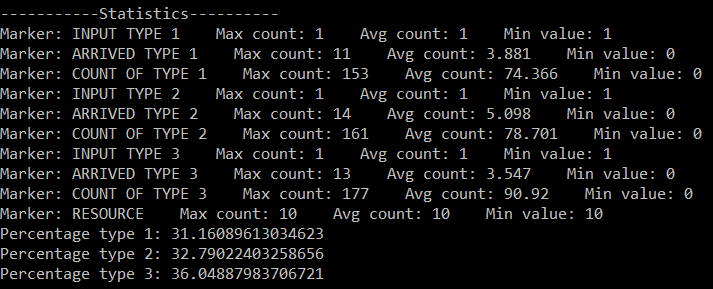


Рисунок 9 – результати роботи алгоритму імітації для завдання 3

**Висновки**

* У завданні 1 із рисунку 2 можна переглянути протокол роботи алгоритму і впевнитися у його правильності проаналізувавши рисунок 3, де виведено результати роботи алгоритму. Як бачимо на рисунку 3 максимальна кількість маркерів у керуючого сигналу “CAN BE SENT” складає 1. Отже, два повідомлення не можуть відправлятися одночасно.
* У завданні 2 із рисунку 5 можна переглянути протокол роботи алгоритму і впевнитися у його правильності проаналізувавши рисунок 6, де виведено результати роботи алгоритму. Як бачимо на рисунку 3 максимальна кількість маркерів в буфері “BUFFER” складає значення n і не перевищує його. Отже, якщо кількість маркерів буфера максимальне, то надходження до нього не відбувається.
* У завданні 3 із рисунку 8 можна переглянути протокол роботи алгоритму і впевнитися у його правильності проаналізувавши рисунок 9, де виведено результати роботи алгоритму. Як бачимо на рисунку 3 процентне співвідношення приблизно однакове, так як нема затримки у переходах PROCESSING TYPE 1, PROCESSING TYPE 2, PROCESSING TYPE 3.