**РОЗДІЛ 2. РОЗРОБКА ПРОГРАМИ ПРГ 1 ДЛЯ ПКС СП**

У даному розділі розроблюється програма ПРГ1 для системи зі спільною пам’яттю, що відповідає технічному завданню, представленому на малюнку нижче.

A = Bi∙max(Z)+ min(Z)∙B∙α(MO∙MK)

**ОП**

0

P-1

…

**2.1 Розробка паралельного математичного алгоритму.**

Паралельний математичний алгоритм відповідно до рекомендованої методичної літератури [10] можна подати у вигляді наступних двох етапів:

1. *minZi = min(ZH)i = ;*

*minZ = min(minZ, minZi), i = ;*

1. *maxZi = max(ZH), i =   
   maxZ = max(maxZ, maxZi), i =*
2. *MAH = MO∙MKH;*
3. *AH = Bi∙maxZi+ minZi∙B∙αiMAH, i = ,*

де:

* H = N/P;
* AH – H рядків вектора А;
* MKH – H рядків матриці MK;
* ZH – H рядків вектора Z;
* MAH – H рядків матриці MA
* MKH – H рядків матриці MK

Спільні ресурси: α, B, min,max, MO,MA

**2.2 Розробка алгоритмів процесів**

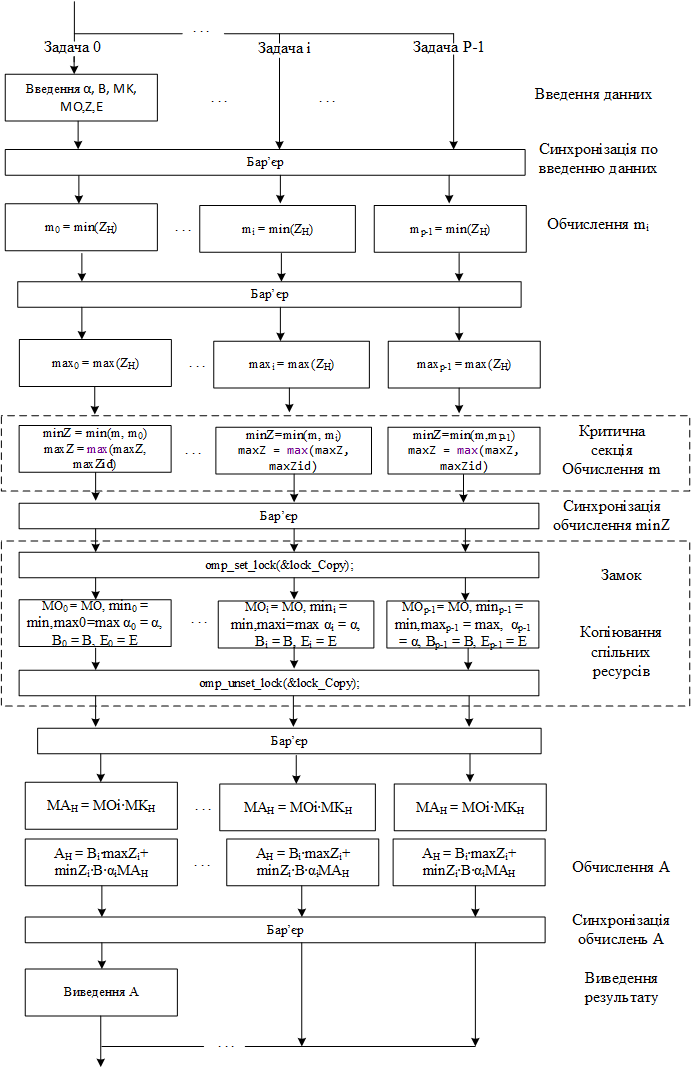
Оскільки розроблюване програмне забезпечення є масштабованим і працює на системі із кількістю процесорів P≥2, то зручним варіантом реалізації є написання єдиного алгоритму для всіх задач.

|  |  |
| --- | --- |
| Задачі Т(0) – Т(P-1) | |
| Крок алгоритму | КД, Бар’єри |
| 1. Якщо tid = 0 Ввести α, B, MK, MO,Z,E. |  |
| 2. **Бар’єр** для усіх задач. Синхронізація по вводу. | Бар’єр |
| 3. Обчислення *minZi = min(Z), i =*  4. Обчислення *maxZi = max(Z), i =* |  |
| 5. Обчислення *minZ = min(minZ, minZi),  maxZ = max(maxZ, maxZi), i =* | КД |
| 6. **Бар’єр** для усіх задач. Синхронізація обчислень minZ,maxZ | Бар’єр |
| 7. **Копіювати** MOi=MO, *minZi = minZ, maxZi = maxZ , αi = α, Bi = B* | КД |
| 8. Обчислення MAH = MOi∙MKH |  |
| 9. **Бар’єр** для усіх задач. Синхронізація обчислень MA | Бар’єр |
| 10. Обчислення *AH = Bi∙maxZi+ minZi∙B∙αiMAH, i =* |  |
| 11. **Бар’єр** для усіх задач. Синхронізація обчислень A | Бар’єр |
| 12. Якщо tid = 0 вивести результат AH. | ­ |

**2.3 Розробка схеми взаємодії процесів**

На основі алгоритму для всіх задач, наведеному в попередньому розділі, було розроблена структурна схема взаємодії задач (рис. 2.1). Вона дозволяє наочно контролювати бар’єри та критичні ділянки. Крім того, на структурній схемі уводяться також замок, та критична секція, що будуть використовуватись в програмі.

Для демонстрації взаємодії між задачами вибрано дві задачі: T(0), T(i) (i = ). Задача T(0) вводить дані, тому з нею взаємодіють всі інші (синхронізація по вводу). Задачі T(i) (i = ) виконують лише обчислення і синхронізуються зі всіма іншими задачами по обчисленню. Задача T(0) виводить результат обчислень.

-Рис. 2.1 Структурна схема взаємодії задач

**2.4 Розробка програми ПРГ1**

Програма написана на мові C++ з використанням бібліотеки OpenMP та складається з трьох модулів: основного СourseWork\_PP2.cpp, та допоміжного operations.cpp і файлу заголовків operations.h.

Основний модуль pro2\_course\_work\_prg1.cpp містить одну функцію:

* main – точка входу в програму. Формує ідентифікатори tid задач, запускає задачі, а також вимірює час виконання elapsedTime програми ПРГ1.

В цьому модулі об’явлена така змінна: P – налаштування кількості процесорів.

Файл заголовків operations.h. представляє заголовки службових функцій (таких як ввід/вивід, копіювання), та типи вектора і матриці. У ньому також об’явлена змінна N що визначає розмірність матриць і векторів.

Допоміжний модуль operations.cpp реалізовує функції об’явлені у файлі заголовків operations.h.

Лістинг програми ПРГ1 наведено у додатку А.