

## Практична робота 5. Аналіз функціонування багатопроцесорних КС різних топологій

**Мета:** Аналіз функціонування та ефективності обчислювальних систем різних топологій.

### Теоретичні відомості

Найбільш важливим аспектом паралельних ОС з локальною пам'яттю є те, як взаємодіють між собою окремі процесори. Конфігурація системи міжпроцесорного зв'язку істотно впливає на порядок підключення ліній зв'язку, що з'єднують окремі процесори. Організація внутрішніх комунікацій ОС називається топологією.

Залежно від того, чи залишається конфігурація взаємозв'язків незмінною (доки виконується відповідне завдання), розрізняють ОС зі статичною та динамічною топологіями. В статичних ОС структура взаємозв'язків фіксована. В ОС із динамічною топологією в процесі обчислень конфігурація взаємозв'язків за допомогою програмних засобів може бути оперативно змінена.

До статичних топологій відносять такі, де між двома вузлами можливий лише один чи декілька фіксованих шляхів, тобто немає комутуючих пристроїв. З можливих критеріїв класифікації статичних ОС обирають їх розмірність. З цих позицій розрізняють:

- одновимірні топології (лінійний масив);
- двовимірні топології (кільце, зірка, дерево, решітка);
- багатовимірні топології (повнозв'язана топологія);
- гіперкубічну топологію.

У простій лінійній топології вузли системи утворюють одновимірний масив та з'єднані у ланцюг (рис. 5.1).



Рисунок 5.1 – Лінійна топологія ОС

У лінійній топології час передавання повідомлення залежить від відстані між вузлами, а відмова одного з них може призвести до неможливості передавання повідомлення. З цього приводу в лінійних ОС використовують відмовостійкі вузли, які при відмові ізолюють себе від мережі, дозволяючи повідомленню обминути несправний вузол.

Стандартна кільцева топологія (рис. 5.2) являє собою лінійний ланцюг, кінці якого з'єднані між собою. Недолік: додавання чи вилучення вузла потребує демонтажу мережі.

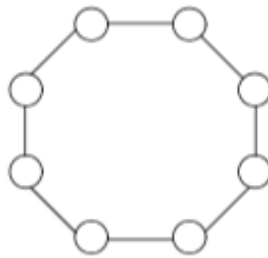


Рисунок 5.2 – Кільцева топологія ОС

Зіркоподібна організація вузлів (рис. 5.3) рідко використовується для об'єднання процесорів багатопроекторної ОС, але добре працює, коли потік інформації йде від декількох вторинних вузлів, з'єднаних з одним первинним вузлом, наприклад при підключенні терміналів. Загальна пропускна здатність мережі звичайно обмежується швидкістю концентратора. Основна перевага зіркоподібної схеми у тому, що конструктивне виконання вузлів на кінцях звичайно може бути дуже простим.

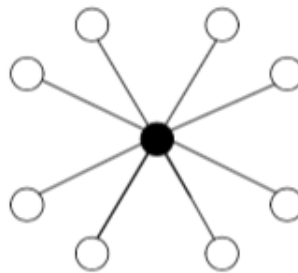


Рисунок 5.3 – Зіркоподібна топологія ОС

У деревоподібній топології (рис. 5.4) система будується за схемою двійкового дерева, де кожний вузол більш високого рівня зв'язаний з двома

вузлами наступного за порядком більш низького рівня. Вузол, що знаходиться на більш високому рівні, називають батьківським, а два підключених до нього нижчерозташованих вузла – дочірніми. В свою чергу, кожний дочірній вузол виступає як батьківський для двох вузлів наступного більш низького рівня. Кожний вузол зв'язаний лише з двома дочірніми та одним батьківським.

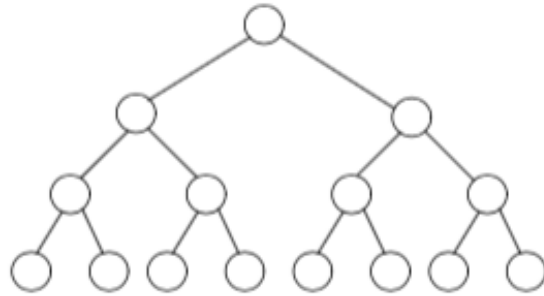


Рисунок 5.4 – Деревоподібна топологія ОС

При великих об'ємах передачі даних між несуміжними вузлами деревоподібна топологія виявляється недостатньо ефективною, оскільки повідомлення повинні проходити через один чи декілька проміжних процесорів. На більш високих рівнях системи ймовірність затору через недостатньо високу пропускну здатність ліній зв'язку вище.

ОС з решітчастою топологією (рис. 5.5) орієнтовані на задачі, пов'язані з обробкою масивів. Їх конфігурація визначається видом та розмірністю масиву.

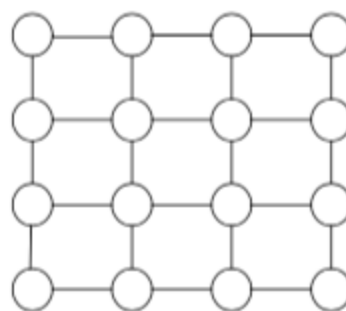


Рисунок 5.5 – Решітчаста топологія ОС

У системах, де кожний процесор з'єднаний з декількома сусідніми процесорами, можна досягти компромісу між складністю системи міжпроцесорного зв'язку, її пропускну здатністю й затримкою обмінів даними. В одновимірних решітках процесорів, що часто називають лінійним масивом,

кожний процесор, крім крайніх, з'єднаний з двома сусідніми. При цьому дані, що пересилаються із процесора-джерела в процесор-приймач, послідовно проходять транзитні процесори, розташовані між джерелом і приймачем. У двовимірній решітці кожний процесор з'єднаний з північним, південним, східним і західним сусідами. Решітки процесорів характеризуються регулярністю, локальністю і простотою маршрутизації міжпроцесорних зв'язків. У тривимірному масиві кожний із процесорів з'єднаний із шістьма сусідами. Перевагою такої системи є мінімальна кількість ліній зв'язку (з кожного процесору виходить не більше двох ліній зв'язку).

У повністюзв'язаній системі кожний процесор має пряме з'єднання з будь-яким іншим процесором.

При об'єднанні паралельних процесорів дуже популярна топологія гіперкуба. Лінія, що з'єднує два вузла (рис. 5.6, а), визначає одновимірний гіперкуб. Квадрат, що утворений чотирма вузлами (рис. 5.6, б) – двовимірний гіперкуб, а куб з 8 вузлів (рис. 5.6, в) – тривимірний гіперкуб та т.ін.

Обмін повідомленнями в гіперкубі базується на двійковому поданні номерів вузлів. Нумерація вузлів робиться так, що для будь-якої пари суміжних вузлів двійкове подання номерів цих вузлів відрізняється лише в одній позиції. Вузли 0010 та 0110 – сусіди, а вузли 0110 та 0101 такими не є.

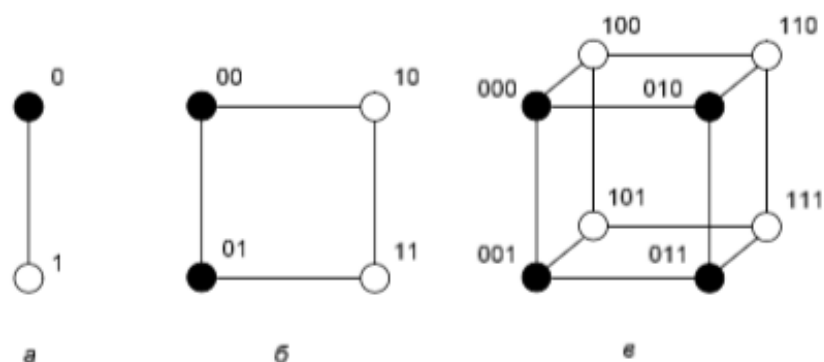


Рисунок 5.6 – Гіперкубова топологія ОС

Така архітектура дає малу кількість зв'язків між процесорами. В ОС з такою архітектурою обчислювальний процес будується таким чином: кожний процесор у вузлі має власну пам'ять та відповідно володіє потужним

обчислювальним ресурсом. Якщо обчислювальної потужності не вистачає, то до вирішення залучають процесори з сусідніх вузлів чи всього кубу. Якщо і цього недостатньо, то залучають процесори, які розташовані у вузлах зовнішнього гіперкубу відносно до даного.

Недоліки окремих типів систем мінімізуються при їх комбінуванні (гібридні системи міжпроцесорного зв'язку). Наприклад, конфігурацію піраміди одержують додаванням зв'язків між процесорами, що належать одному ярусу дерева, відповідно до конфігурації двовимірних решіток. Таким чином, у піраміді об'єднані переваги дерева й решітки.

Структури ОС із кластерами також відносять до гібридних систем зв'язку. У межах кластера процесори з'єднані відповідно до однієї з конфігурацій зв'язків, наприклад, загальною шиною, а кілька кластерів об'єднані у ОС за допомогою іншої конфігурації зв'язків, наприклад, у вигляді решітки або гіперкуба.

У динамічній топології ОС з'єднання вузлів забезпечується електронними ключами, варіюючи установки яких можна змінювати топологію системи. У вузлах динамічних ОС розташовуються комутуючі елементи, а пристрої, що обмінюються повідомленнями (термінали), підключаються до входів та виходів цієї мережі. Як термінали можуть виступати процесори чи процесори та модулі пам'яті. Для таких ОС частіше всього використовується одно- або багатокаскадна комутація на основі матричних комутаторів.

ОС з шинною архітектурою – найбільш простий та дешевий вид динамічних ОС. При одношинній топології (рис. 5.7), усі вузли підключені до однієї шини, що сумісно використовується. В кожний момент часу обмін повідомленнями може вести лише одна пара вузлів, тобто на період передавання повідомлення шину можна розглядати як мережу, що складається з двох вузлів.

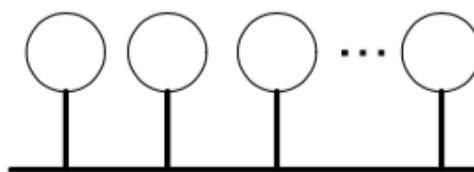


Рисунок 5.7 – Одношинна топологія ОС

Більш ефективною архітектурою динамічних ОС є система, в якій процесори зв'язані між собою за допомогою матричного комутатора. У цьому випадку в кожний момент часу обмін повідомленнями можуть вести  $n/2$ - пар вузлів, де  $n$  – кількість вузлів у системі. Недоліком такої архітектури є висока вартість.

Приклад: Побудувати часову діаграму роботи ОС для трипроцесорної ОС топології “Зірка” для заданного на рис. 5.8. дерева обчислювальної задачі:

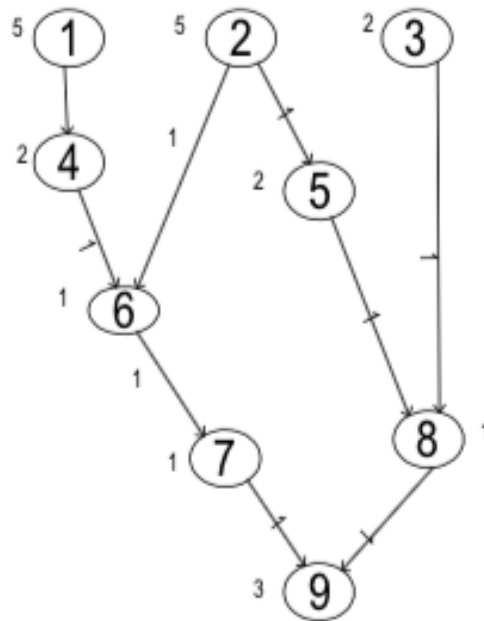


Рисунок 5.8 – Дерево обчислювальної задачі

ОС для трьох процесорів топології “Зірка” подана на рис. 5.9.

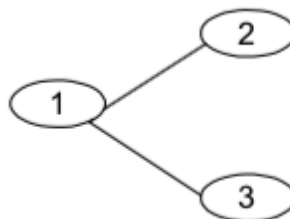


Рисунок 5.9 – Топологія “Зірка”

Часову діаграму роботи ОС для трипроцесорної ОС топології “Зірка” наведено на рис. 5.10.

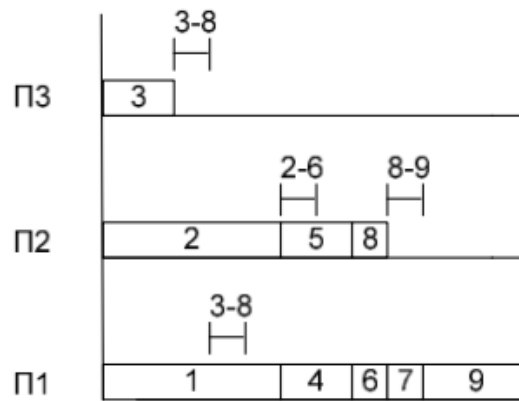


Рисунок 5.10 – Часова діаграма роботи ОС для трьохпроцесорної ОС топології “Зірка”

Таблиця 5.1 - Час виконання операції в ОС

Час Но- мер ва- ріан- та	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	4	2	6	1	3	2	7	1	1	3	6	3	2	3	6	5	1	1	4
1	1	3	3	5	2	1	3	4	3	2	2	2	8	2	1	3	2	5	4
2	1	4	6	5	5	3	2	2	3	3	1	7	5	1	2	6	1	2	3
3	1	4	4	3	2	5	6	2	1	1	3	4	1	4	2	5	7	3	2
4	2	2	4	2	5	3	2	1	5	3	2	7	3	6	2	7	3	1	1
5	3	4	3	8	4	2	6	4	5	3	3	7	4	5	2	4	2	3	1
6	4	6	1	4	3	5	4	3	6	3	2	4	3	4	6	2	5	7	8
7	1	5	2	3	2	7	2	1	6	3	4	5	6	1	2	6	4	4	2
8	7	2	5	2	1	3	2	2	5	3	4	3	4	2	3	7	2	5	2
9	5	2	6	1	5	3	4	4	3	2	1	2	8	7	5	2	1	3	3

Вихідні дані:

- а) час виконання операцій (табл. 5.1.);
- б) топологія ОС та кількість процесорів (табл. 5.2).

Час виконання операцій визначають останньою цифрою залікової книжки. Типи топологій та кількість процесорів для аналізу ОС визначають за передостанньою цифрою залікової книжки.

## Хід роботи

1. Вивчити теоретичні відомості.
2. Отримати допуск до роботи.
3. Вибрати за методичними вказівками варіант завдання.
4. Використовуючи дані з практичної роботи 4 (рис. 4.6. – 4.10), побудувати часові діаграми роботи ОС заданих топологій та кількості процесорів (табл. 5.2) з заданим часом виконання операцій (табл. 5.1).
5. Визначити час виконання обчислювальної задачі для кожної топології, що аналізується.
6. Скласти звіт по лабораторній роботі
7. Зробити висновки по роботі, визначивши, яка з топологій найбільш оптимальна для вирішення конкретної обчислювальної задачі.

Таблиця 5.2 – Топологія ОС та кількість процесорів

Тип топології	Лінійка			Кільце			Решітка		Зірка			Дерево		Гіперкуб	
	2	3	4	2	3	4	4	9	2	3	4	3	7	4	8
0							*	*	*	*	*			*	*
1	*	*	*	*	*	*						*	*		
2									*	*	*	*	*	*	*
3	*	*	*						*	*	*	*	*		
4	*	*	*	*	*	*	*	*							
5				*	*	*	*	*				*	*		
6	*	*	*				*	*						*	*
7				*	*	*			*	*	*			*	*
8	*	*	*						*	*	*			*	*
9				*	*	*	*	*				*	*		