

Міністерство освіти і науки України  
Національний авіаційний університет  
Навчально-науковий інститут комп'ютерних інформаційних технологій  
Кафедра комп'ютеризованих систем управління

Лабораторна робота №3  
з дисципліни «Комп'ютерні системи»  
на тему «Вивчення роботи багатопроцесорних обчислювальних систем»  
Варіант №3

Виконав:  
студент ННІКІТ  
групи СП-325  
Клокун В. Д.  
Перевірив:  
Ковальов М. О.

Київ 2019

## 1. МЕТА РОБОТИ

Закріплення теоретичних знань зі структури багатопроцесорних ОС.

## 2. ЗАГАЛЬНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Передумовою для створення обчислювальної системи є мікромініатюризація компонентів схем (інтегральна технологія), підвищення їх надійності, зниження вартості та енергоспоживання. Все це дозволяє об'єднати в одному апаратурному комплексі більше елементів, ніж у звичайних ЕОМ, і одержати збільшення швидкодії не тільки за рахунок підвищення швидкостей окремих елементів і скорочення довжин з'єднань, але й за рахунок ускладнення структури системи.

Багатопроцесорні обчислювальні системи можна розділити на багатопроцесорні обчислювальні системи з роздільним керуванням (тип М1), з загальним керуванням (тип М2), та матричні обчислювальні системи (тип М3).

На рис. 1 зображена багатопроцесорна обчислювальна система типу М1 із роздільним керуванням. Система вміщує  $N$  процесорів, кожен з яких має свій пристрій керування, арифметико-логічний пристрій та пристрій пам'яті. Всі процесори зв'язані між собою через загальну пам'ять і з нею ж зв'язані спільні для всіх процесорів канали зовнішнього обміну. Загальна пам'ять — це частина внутрішньої пам'яті системи. Можливо, що пам'ять в складі кожного процесора відсутня або являє собою надоперативну пам'ять. Тоді відповідно загальне поле пам'яті містить або всі ступені внутрішньої пам'яті, або головну пам'ять і велику пам'ять, або тільки велику пам'ять.

Можливі і такі побудови обчислювальних систем типу М1, в яких загальної внутрішньої пам'яті взагалі немає, а обмін інформації йде через окремі ланцюги. При цьому загальні канали адресуються до всієї внутрішньої пам'яті системи, указуючи як адресу звернення, номер процесора та адресу пам'яті.

Обчислювальні системи типу М1 називають системами з множинним потоком команд і множинним потоком даних (МКМД). Такі системи краще всього пристосовані до використання паралелізму незалежних гілок. Суть цього виду паралелізму полягає в тому, що в програмі розв'язання великої задачі на окремих етапах можна виділити незалежні частини гілки, які можуть виконуватися паралельно (одночасно одна з одною).

Для системи М1 виконання гілки доручається відповідному процесору і можливості обчислювальної системи будуть використані найкращим чином, якщо число незалежних гілок задачі (програми) буде точно дорівнювати або буде кратним числу процесорів у системі.

Окрім того, важливо, щоб усі гілки програми, які входять до одного й того ж ярусу, були б по можливості однакової довжини (трудомісткості) та щоб

ця довжина була б достатньо великою.

Кратність ширини ярусів графа кількості процесорів може бути отримана тільки в разі збігу майже неймовірних обставин.

На таку ситуацію можна розраховувати або у спеціалізованій системі, яка побудована для розв'язання певної задачі (класу задач), або як на результат спеціального перетворення програми розв'язання задачі.

Тому ОС типу M1 з великою кількістю процесорів використовуються для досягнення високої системної продуктивності, тобто продуктивності, яка досягається при розв'язанні сукупності задач користувача.

Основною позитивною якістю даних ОС є потенційно дуже велика взаємодія процесорів. Великий недолік полягає у тому, що загальна пам'ять може бути потрібна одночасно різним процесорам. У таких випадках виникають затримки з доступом до пам'яті: тривалість подібних затримок, яка називається часом конфліктів пам'яті, може зростати зі збільшенням кількості процесорів.

На рис. 2 наведено можливий варіант обчислювальної системи типу M2 із загальним керуванням, яка орієнтована на паралелізм суміжних операцій.

Під загальним керуванням зібрано: пристрій управління, N арифметико-логічних процесорів, кожний з них може мати свою невелику пам'ять, але може і не мати її. В основному процесори працюють над загальною пам'яттю, пристроєм керування якої вважаються інструкції.

Є два принципових варіанти організації роботи такої системи. Перший варіант (тип M2.1.) полягає в тому, що пристрій керування зчитує з пам'яті інструкції «вперед» зі швидкістю, яка набагато перевищує швидкість виконання однієї інструкції одним процесором; прочитавши інструкцію, аналізує, чи є умови для того, щоб почати її виконання, а якщо є, то доручає її виконання будь-якому вільному процесору (але першому, який вивільниться, якщо всі вони були зайняті) та аналізує наступну інструкцію і т. ін.

Перегляд програми «вперед» та виконання окремих її інструкцій може продовжуватись і у тому випадку, коли з'ясовується, що виконання чергової інструкції потрібно відкласти. Інструкції, виконання яких відкладено, накопичуються у буфері пристрою керування.

В другому варіанті (тип M2.2.) пристрій керування зчитує з пам'яті безпосередньо вектори-інструкції (VLIW-структури ОС), один вектор-інструкція містить N компонентів — за числом процесорів у системі. Кожна з них вказує, яку операцію повинен виконувати відповідний процесор.

Для ОС типу M2.1. аналіз програми і організація паралельних обчислень проводиться пристроєм керування безпосередньо на етапі виконання програми (методом інтерпретації), для ОС типу M2.2. — на етапі програмування під час трансляції програми. Зрозуміло, що оптимізація програми методом трансляції може бути виконана більш ефективно, ніж інтерпретація. Окрім того, при виконанні програми в ОС типу M2.1. додатковий час буде витратитись на ана-

ліз програми.

Недоліком обчислювальної системи типу M2.2. є залежність часу виконання будь-якого вектора-інструкції від тривалості виконання самої тривалої операції в ньому.

Обчислювальні системи типу M2 орієнтовані на паралелізм суміжних операцій. Його суть полягає в наступному. Під час виконання програми зустрічаються ситуації, коли вихідні дані для деякої  $i$ -ї операції формуються не під час виконання попередньої  $(i - 1)$ -ї операції, а раніше — під час виконання, наприклад  $(i - 2)$ -ї чи  $(i - 3)$ -ї операції. Тоді можна сполучити час виконання  $i$ -ї операції з виконанням  $(i - 2)$ -ї чи  $(i - 3)$ -ї операції.

Кількість процесорів у системах типу M2 не робиться більше ніж 2–6, оскільки глибина паралелізму суміжних операцій звичайно невелика.

Характерною особливістю систем типу M2 є той факт, що, незважаючи на наявність загального для усіх  $N$  процесорів пристрою керування, усі вони управляються різними інструкціями чи різними компонентами вектора-інструкції.

Системи типу M2 висувають певні вимоги до пам'яті. У першому варіанті вимагається висока швидкодія пам'яті (щоб читати інструкції швидше, ніж йде їх виконання), у другому вимагається широкий формат звернень до пам'яті (щоб за одно звернення прочитати один вектор-інструкцію).

«Висока швидкодія» пам'яті — це час звернення до пам'яті, приблизно в  $N$  разів менше, ніж час виконання операції у процесорі; «широкий формат» пам'яті — це можливість вибірки орієнтовно  $N$  слів за одно звертання.

Принцип побудови обчислювальних систем типу M3 наведено на рис 3. Загальний пристрій керування роздає всім  $N$  процесорам однакові команди для кожного кроку. Всі процесори виконують одну і ту ж операцію — кожен над своїми даними. Системи типу M3 називають матричними системами (або з матричним процесором), а самі процесори — процесорними елементами матричного процесора.

Число процесорних елементів у матричному процесорі може досягати значної кількості.

Матричні обчислювальні системи орієнтовані на розв'язання задач з природним паралелізмом. Такі задачі зводяться до операцій над багатомірними векторами, матрицями та іншими аналогічними об'єктами. Кожен з цих об'єктів може бути поданий сукупністю чисел: вектора — своїми компонентами, матриць — своїми елементами. Більшість операцій при цьому — сукупності однакових операцій над відповідними парами чисел (елементами) двох аналогічних об'єктів. Наприклад, складання двох векторів полягає в додаванні відповідних компонент векторів. Очевидно, що всі ці операції складання можуть бути виконані паралельно і незалежно одна від одної.

### 3. ХІД РОБОТИ

Вихідними даними для лабораторної роботи є арифметичний вираз. Для варіанту №3 він виглядає так:

$$(A + B) / (C + D \cdot E) + F + (G + H) / K \cdot L. \quad (1)$$

За даним виразом складаємо дерево арифметичного виразу (рис. 1).

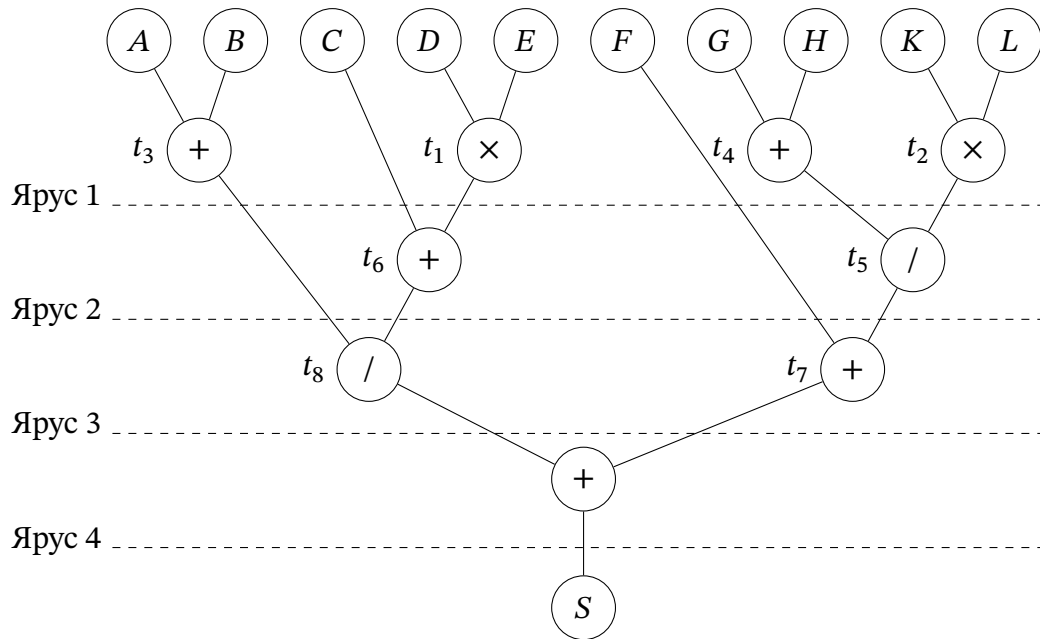


Рис. 1: Дерево арифметичного виразу  $(A + B) / (C + D \cdot E) + F + (G + H) / K \cdot L$

### 4. ВИСНОВОК

Виконуючи дану лабораторну роботу, ми закріпили теоретичні знання зі структури багатопроцесорних ОС.