

ПАРАЛЕЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ

В умовах постійно зростаючих вимог до продуктивності обчислювальної техніки все більш явними стають обмеження класичної фон-нейманівської архітектури. Подальший розвиток обчислювальної техніки зв'язаний з переходом до паралельних обчислень як у рамках однієї обчислювальної машини, так і шляхом створення багатопроцесорних систем і мереж, які об'єднують велику кількість окремих процесорів або окремих обчислювальних машин. Для такого підходу замість терміну «обчислювальна машина» більш підходить термін «обчислювальна система» (ОС). Головною особливістю такої системи є наявність у ній засобів, які реалізують паралельну обробку інформації.

1 РІВНІ ПАРАЛЕЛІЗМУ

Паралельна обробка інформації являє собою одночасне рішення двох або більшої кількості частин однієї й тієї ж програми двома чи більшою кількістю ЕОМ (процесорними елементами) обчислювальної системи. Реалізують три основних способи організації паралельної обробки:

1) суміщення у часі різноманітних етапів різних задач – це мульти-програмна обробка, яка широко використовується як у однопроцесорних ЕОМ, так і в складних ОС;

2) одночасне розв'язання різноманітних задач або частин однієї задачі (можливо тільки за наявності декількох обробляючих пристроїв);

3) конвеєрна обробка інформації.

Перші два способи використовують особливості паралельних задач або потоків задач, що дозволяє здійснювати той або інший паралелізм. Перший тип паралелізму – це *природний паралелізм незалежних задач*, який полягає у тому, що в систему поступає безперервний потік не зв'язаних між собою задач. У цьому випадку розв'язання будь-якої задачі не залежить від результатів розв'язання інших задач, що дозволяє підвищити продуктивність ОС у разі використання декількох обробляючих пристроїв.

Одним з найбільш розповсюджених типів паралелізму є *паралелізм незалежних гілок*. Суть його полягає у виділенні окремих незалежних частин великої задачі (гілок програми), які можуть виконуватись паралельно окремими обробляючими пристроями незалежно один від одного. Причому обробляючі пристрої ОС функціонують в однопрограмному режимі паралельної обробки інформації. Двома незалежними гілками програми визнаються гілки програми, що відповідають таким умовам:

- ні одна з вхідних для гілки програми величин не є вихідною величиною іншої програми (відсутність функціональних зв'язків);
- для двох гілок програми не повинен здійснюватись запис у одні й ті ж комірки пам'яті (відсутність зв'язку по оперативній пам'яті);
- умови виконання однієї гілки не залежать від результатів, що отримані під час виконання іншої гілки (незалежність по управлінню);
- обидві гілки повинні виконуватись у різних блоках програми (програмна незалежність).

Виділення незалежних гілок широко використовується під час паралельної обробки в задачах матричної алгебри, лінійного програмування, спектральної обробки сигналів, прямого та зворотного перетворення Фур'є та ін.

Методи та засоби реалізації паралелізму залежать від того, на якому рівні він повинен забезпечуватись. Зазвичай розрізняють такі *рівні паралелізму*:

- **рівень завдань** – декілька незалежних завдань одночасно виконуються на різних процесорах, які практично не взаємодіють один з одним. Цей рівень реалізується в ОС з множиною процесорів у багатозадачному режимі.

- **рівень програм** – частини однієї задачі виконуються множиною процесорів. Даний рівень досягається на паралельних ОС.

- **рівень команд** – виконання команди розділяється на фази, а фази декількох послідовних команд можуть бути перекриті за рахунок конвеєризації. Даний рівень використовується в ОС з одним процесором.

- **рівень бітів (арифметичний рівень)** – біти слова обробляються один за одним. Цей процес має назву *біт-послідовна операція*. Якщо біти слова обробляються одночасно, кажуть про *біт-паралельну операцію*. Даний рівень реалізується в звичайних і суперскалярних процесорах.

Паралелізм рівня завдання можливий між незалежними завданнями або їх фазами. Основним засобом реалізації паралелізму на рівні завдань є багатопроцесорні і багатомашинні обчислювальні системи, в яких завдання розподіляються за окремими процесорами або машинами. Однак, якщо кожне завдання трактувати як сукупність незалежних задач, реалізація даного рівня можлива і в рамках однопроцесорної ОС. У цьому випадку декілька завдань можуть одночасно знаходитись в основній пам'яті ОС, за умови, що в кожному момент виконується тільки одне з них. Коли завдання, яке виконується, потребує вводу/виводу, ця операція запускається, а до її завершення інші ресурси ОС передаються другому завданню. По завершенні вводу/виводу ресурси ОС повертаються до завдання, яке ініціювало цю операцію. В цьому випадку паралелізм забезпечується за рахунок того, що центральний процесор і система вводу/виводу функціонують одночасно і обслуговують різні завдання.

Паралелізм виникає також, коли у незалежних завдань, які виконуються в ОС, є декілька фаз, наприклад обчислення, запис у графічний буфер, системні виклики. За те, як різні завдання впорядковуються і витрачають загальні ресурси відповідає операційна система.

Паралелізм рівня програм. Про паралелізм на рівні програми можна говорити у двох випадках. По перше, коли в програмі можна виділити незалежні ділянки, які допустимо виконувати паралельно. Другий тип паралелізму програм можливий у межах окремого програмного циклу, якщо в ньому окремі ітерації не залежать один від одного. Програмний паралелізм можна реалізувати за рахунок великої кількості процесорів або множини функціональних блоків.

Загальна форма паралелізму на рівні програм організується розбиттям даних, які програмуються на підмножини. Цей розподіл називають *декомпозицією області* (domain decomposition), а паралелізм, який при цьому виникає, має назву *паралелізму даних*. Підмножини даних призначаються різним обчислювальним процесам, і цей процес має назву *розподілення даних* (data distribution). Процесори виділяються певним процесам або за ініціативою програми, або в процесі роботи операційною системою. На кожному процесорі може виконуватись більше одного процесу.

Паралелізм рівня команд. Паралелізм на рівні команд має місце, коли обробка декількох команд або виконання різних етапів однієї і тієї ж команди може перекриватись у часі. Розробники обчислювальної техніки ще здавна зверталися до методів, відомих під загальною назвою «поєднання операцій», при якому апаратура ОМ у будь-який момент часу виконує одночасно більше однієї операції. Цей загальний принцип включає два поняття: *паралелізм* і *конвеєризацію*. Хоча у них багато загального і їх часто важко розрізнити на практиці, ці терміни відображають два принципово різних підходи.

У першому варіанті поєднання операцій досягається за рахунок того, що в складі обчислювальної системи окремі пристрої присутні в декількох копіях. Так, до складу процесора може входити декілька АЛП, і висока продуктивність забезпечується за рахунок одночасної роботи всіх цих АЛП. Другий підхід був розглянутий раніше.

Під час організації паралельного обчислювального процесу виникає задача вибору побудови розкладу.

Розклад паралельних обчислювальних процесів визначає порядок виконання програми в ОС, включаючи розподіл частин програми по обробляючих пристроях (процесорах, ОМ), і служить основою алгоритмів планувальника операційної системи та різноманітних управляючих програм. Як критерії оптимальних розкладів для паралельної програми можна назвати:

- мінімізацію часу виконання програми;
- мінімізацію кількості потрібних пристроїв обробки;
- мінімізацію середнього часу закінчення виконання завдань;
- максимізацію завантаження пристроїв ОС;
- мінімізацію часу простоювання пристроїв.

Найчастіше використовують перший критерій.

2 КЛАСИФІКАЦІЯ АРХІТЕКТУР КОМП'ЮТЕРНИХ СИСТЕМ

Існує декілька класифікацій архітектур комп'ютерних систем. Найбільш вдалою є класифікація М. Флінна.

Архітектурні особливості комп'ютерних систем описано з погляду потоку команд (інструкцій) та потоку даних. Такий підхід дає змогу відносити архітектури комп'ютерів до одного з чотирьох класів (табл. 6.1, рис. 6.1).

Класифікація здійснена з погляду не структури, а того, як у комп'ютері його машинні команди взаємодіють з даними.

Таблиця 1 - Класифікація архітектур комп'ютерних систем

Потік команд	Одиничний потік даних (ОД)	Множинний потік даних (МД)
Одиничний (ОК)	ОКОД (SISD) (однопроцесорні комп'ютери)	ОКМД (SIMD) (комп'ютери з паралельними або асоціативними процесорами)
Множинний (МК)	МКОД (MISD) (конвеєрні магістральні комп'ютери)	МКМД (MIMD) (багатопроцесорні або багатомашинні комплекси)

SISD (Single Instruction Stream/Single Data Stream) - одиничний потік команд і одиничний потік даних. Представник цього класу – класичний фон-нейманівський комп'ютер. Команди обробляються послідовно і кожна команда ініціює одну операцію з одним потоком даних. До цього класу комп'ютерів можна віднести також конвеєрні комп'ютери. Деякі спеціалісти вважають, що до SISD-систем можна віднести і векторно-конвеєрні ОС, якщо розглядати вектор як неподільний елемент даних для відповідної команди.

MISD (Multiple Instruction Stream/Single Data Stream) – множинний потік команд і одиничний потік даних. В архітектурі присутня множина процесорів, які обробляють один і той же потік даних. Ряд дослідників відносять до даного класу конвеєрні системи. Прийнято вважати, що доки даний клас незадіяний, однак може бути корисним для розробки принципово нових концепцій побудови обчислювальних систем [25].

SIMD (Single Instruction Stream/Multiple Data Stream) – одиничний потік команд і множинний потік даних. Ця архітектура дозволяє виконати одну арифметичну операцію відразу над багатьма даними – елементами вектору. Представниками цього класу є системи з матрицею процесорів, де один управляючий пристрій контролює множину процесорних елементів. Усі процесорні елементи отримують від пристрою управління однакову команду і виконують її над власними локальними даними. В цей клас можуть бути включеними і векторно-конвеєрні ОС, якщо кожний елемент вектора розглядати як окремий елемент даних.

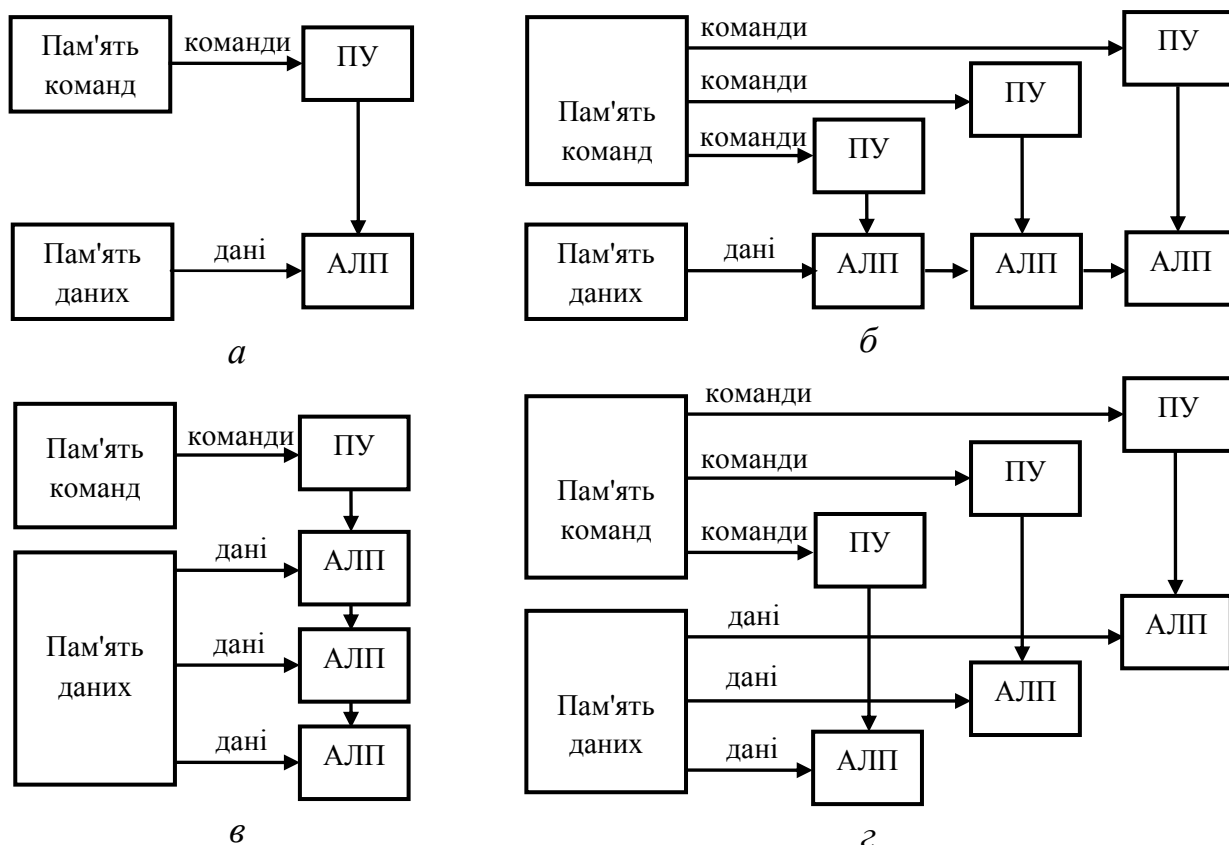


Рисунок 1 - Архітектура комп'ютерних систем за Флінном:
а - SISD; *б* - MISD; *в* - SIMD; *г* - MIMD

MIMD (Multiple Instruction Stream/Multiple Data Stream) – множинний потік команд і множинний потік даних. До цього класу відносяться системи з множиною пристроїв обробки команд, які об'єднані в єдиний комплекс і кожний працює з власним потоком команд. Цей клас надзвичайно широкий, бо включає в себе різного роду мультипроцесорні системи.

Класифікація Флінна надто загальна, вона має деякі недоліки, наприклад відносить усі паралельні комп'ютери, крім мультипроцесорних, до одного класу і не вказує ніякої відмінності між конвеєрними комп'ютерами і матрицею МП.

Використовуються й інші класифікації архітектур, зокрема систематика Ф. Шара, структурна систематика Р. Хокні та К. Джессхоупа.

Структурна систематика Р. Хокні та К. Джессхоупа

На першому рівні всі обчислювальні системи поділяються за принципом множинності (кількості) на однокомп'ютерні та багатокомп'ютерні. Обчислювальні системи з одним комп'ютером, у свою чергу, поділяються на ОМ з одним конвеєрним МП та з багатьма МП.

Перші з них є традиційними послідовними комп'ютерами, а другі утворюють клас паралельних комп'ютерів, які поділяються на конвеєрні, неконвеєрні та мікропроцесорні матриці.

Прикладом однієї з перших неконвеєрних обчислювальних машин з паралелізмом є комп'ютер CDC – 6600, побудований на основі декількох скалярних процесорів.

Конвеєрні ЕОМ поділяються на такі, що виконують тільки скалярні команди, наприклад комп'ютери CDC – 7800, FPC AP-120B, і такі, що виконують векторні команди. Комп'ютери, що використовують векторні команди, поділяють, у свою чергу, на комп'ютери із спеціалізованим конвеєром, наприклад CRAY-1, та з універсальним конвеєром – комп'ютер CYBER 205.

Комп'ютери з класу машин з матрицею процесорів поділяють за зв'язаністю процесорів у матриці, розрядністю тощо. Першими машинами такого типу були ILLIAC-IV, BSP, STA-RAN, ICL DAP, OMEN та ін.